Journal of Mobile Multimedia, Vol. 6, No. 3 (2010) 263–280

# SITUATION AWARE COGNITIVE ASSISTANCE IN SMART HOMES

LIMING CHEN CHRIS NUGENT
*School of Computing and Mathematics, University of Ulster*
*Shore Road, Newtownabbey, County Antrim, BT37 0FS, Northern Ireland*
*l.chen; cd.nugent@ulster.ac.uk*

Received September 16, 2009
Revised May 2, 2010

Smart Homes (SH) have emerged as a realistically viable solution capable of providing technology-driven assistive living for the elderly and disabled. Nevertheless, it still remains a challenge to provide situation-aware cognitive assistance for those in need in their Activity of Daily Living (ADL). This paper introduces a systematic approach to providing situation-aware ADL assistances in a smart home environment. The approach makes use of semantic technologies for sensor data modeling, fusion and management, thus creating machine understandable and processable situational data. It exploits intelligent agents for interpreting and reasoning semantic situational (meta)data to enhance situation-aware decision support for cognitive assistance. We analyze the nature and issues of SH-based healthcare for cognitively deficient inhabitants. We discuss the ways in which semantic technologies enhance situation comprehension. We describe a cognitive agent for realizing high-level cognitive capabilities such as prediction and explanation. We outline the implementation of a prototype assistive system and illustrate the proposed approach through simulated and real-time ADL assistance scenarios in the context of situation aware assistive living.

*Keywords*: Ontologies, situation awareness, assistive agent, smart homes, cognitive assistance
*Communicated by*: D. Taniar

## 1 Introduction

With the advance and prevalence of low-cost low-power sensors, computing devices and wireless communication networks, pervasive computing [1] has evolved from a vision to a realistically achievable and deployable computing paradigm. Research is now being conducted in all related areas, ranging from low-level data collection, intermediate-level information processing, to high-level applications and service delivery. It is becoming increasingly evident that the prevalence of intelligent environments to work and live within which flexible multimodal interactions, proactive service provisioning, and situation aware personalized activity assistance, will be commonplace.

As the ever growing ageing population increasingly over-stretches limited healthcare resources, the provision of healthcare is undergoing a fundamental shift towards the exploitation of pervasive computing technologies to support independent living. SH has emerged as one of the mainstream approaches to providing ADL assistances for the elderly, in particular those suffering from cognitive deficiencies such as Alzheimer's disease [2, 3]. A SH is an augmented environment equipped with sensors, actuators, devices and information processing components, inhabited by the elderly or disabled. The rationale is that assistive systems, e.g., an assistive agent, can monitor environmental events and user's behavior through sensors, process and respond timely through actuators or health services, e.g., audio/video outputs or care professionals, to advise the inhabitant the most suitable actions based on the dynamic situation and the inhabitant's ADL profiles.

Existing research has currently concentrated on sensor networks, data collection and communication, and low-level ad hoc responsive assistances based on the simple processing of low-level raw sensor data. For example, if a room temperature is lower than a specific value, the air conditioner will start. Even though existing SH technologies are able to generate massive amounts of data from sensors and mobile devices around the people and entities, it still remains a challenge to provide just-in-time behavioral and cognitive assistance for cognitively deficient inhabitants such as dementia patients who often get lost during their ADL due to bad memory and/or cognitive problems. For instance, to remind a dementia patient to add milk to a cup after a tea bag and hot water have been added. To achieve this, assistive systems have to be able to observe, interpret and reason the dynamic situations in a SH, both temporally and spatially. In other words, assistive systems should have cognitive capabilities to compensate the loss of the inhabitants' cognition capabilities and to guide the inhabitant's behaviour as normal care providers can do. This further requires that the situational data of a smart home be interpretable and processable by assistive systems.

We contend that semantic technologies hold the key to enhanced situation awareness and the potential of SHs can only be fully realized when sensor data are imbued with rich metadata and well-defined meaning. In this paper we propose a semantic-enabled agent-based approach to situation-aware cognitive ADL assistance in a SH. The approach uses semantic technologies for sensor data modeling, fusion and management that generate machine understandable and processable situational data. Semantic data facilitate not only data interoperability, sharing and integration but also high-level automation and advanced processing capabilities. This allows assistive systems (such as software agents) to carry out automated interpretation and reasoning by exploiting semantic situational (meta)data, thus realizing situation-aware ADL assistances.

The paper is organised as follows: Section 2 discusses related work. Section 3 introduces situation awareness and a system architecture for the proposed approach. Section 4 describes semantic data management for enhanced situation awareness. Section 5 presents a cognitive assistive agent for situation interpretation and reasoning. Section 6 outlines a prototype assistive system and illustrates our approach in a real world use scenario. We conclude the paper and point out future work in Section 7.

## 2 Related Work

Making computer systems adaptable to the changes of their operating environments has been previously researched in the context of agent technologies [4]. An intelligent agent is a software system operating in an environment. It senses the changes of the environment, makes a plan in terms of its goal and domain knowledge and takes actions accordingly. An intelligent agent can respond to changes of the environment it inhabits in a number of ways, notably reactive, proactive and adaptive.

Recently technology advances in pervasive computing and ambient intelligence have provoked considerable interest in context-aware applications [5, 6, 7, 8, 9]. Context awareness in pervasive computing refers to a general class of software systems that can sense their physical environments, i.e., their context of use, and adapt their behavior accordingly. Here contextual information mainly consists of location, time, the entities the system interacts with and the surrounding events and resources. However, context awareness and situation awareness have different research focuses. The former is mainly concerned with linking changes in the environments with software systems. The latter rather concentrates on the knowledge and understanding of the environment that is critical to decision making. Situation awareness pays particular attention on the mental model and cognitive processes from the system's perspective.

Some recent and ongoing work on context aware assistive technologies has adopted an ontology based approach [10, 11, 12, 13]. Nevertheless, ontologies are primarily treated as data models for data/service integration, exchange and sharing in these practices. In contrast, our work uses ontologies as conceptual level knowledge models to support automated situational data interpretation and reasoning.

The use of semantic technologies for situation awareness has been studied in military operational context [14, 15, 16]. While our research shares consensus with these endeavours in using ontologies as the situational data models, the fundamental difference is on how such semantic situational data are used. They have concentrated on semantically enabled data fusion and retrieval. Our work focuses on the innovative exploitation of semantic situational data for the provision of high level cognitive capabilities with the purpose of delivering cognitive assistance for SH patients. As such we have introduced an agent based approach to automated situational data comprehension and reasoning. The synergy of semantically enhanced situation awareness with intelligent agents for cognitive ADL assistance has not been seen so far in related research communities.

## 3 A Systematic Approach

A situation is often conceptualized as a snapshot of states at a specific time point in a physical or conceptual environment. Situation awareness has been referred to as "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [17, 18]. From this definition we can figure out that situation awareness is a cognitive process that consists of three operational functions. Firstly, it involves the sensing and recognition of different elements in the environment as well as their characteristics and behaviors. Secondly, it needs the interpretation and comprehension of the significance associated with perceived elements in the environment. And thirdly it requires the ability to anticipate the actions of elements and predict future states of the environment. For entities, either human beings or robots or software systems operating in complex, dynamic and uncertain environments, situation awareness is the determinant of making informed right decisions at the right time in the right place.

Human beings with normal cognitive capabilities are situation aware when they make decisions in performing their activities. Nevertheless, SH inhabitants, in particular those suffering from cognitive deficiencies such as Alzheimer's disease, are incapable of doing this.

As such a basic requirement of assistive systems is that they should be situation aware. Current SH infrastructure has provided sensor networks for perception, but the interpretation and understanding of perceived data and the realization of high-level cognitive capabilities such as prediction, explanation and planning are still missing.

We propose a semantic enabled systematic approach to enhanced situation awareness for assistive systems, as shown in Figure 1. The approach is grounded on three technological pillars, corresponding to the realization of the three operational functions for situation awareness respectively. The first technological underpinning is based on sensor, device and actuator networks that are responsible for monitoring and collecting contextual data. They are mainly embedded in a SH physical environment - as shown in the Smart Home component in Figure 1. The second pillar is semantic modeling, representation and management for a SH as shown in the Semantic Management component, which includes sensor data, situations, ADLs and an inhabitant's ADL profiles. The use of ontologies for data modeling and representation serves two purposes: Firstly it provides a formal way to model and represent interrelations between contextual data from multiple sources, thus facilitating data fusion and construction of situations. Secondly, it gives data rich metadata and well-defined meaning, thus enabling automated comprehension of the significance of contextual data. The third technological pillar is intelligent Assistive Agent that provides high level cognitive capabilities such as prediction, explanation and planning based on reasoning and manipulation of semantic situational data and knowledge. Given the considerable existing work on the physical aspects of SH such as sensors and underlying communication networks, we focus on semantic data management for enhanced situation awareness and assistive agent for the realization of cognitive activity assistance, which are described in details below.

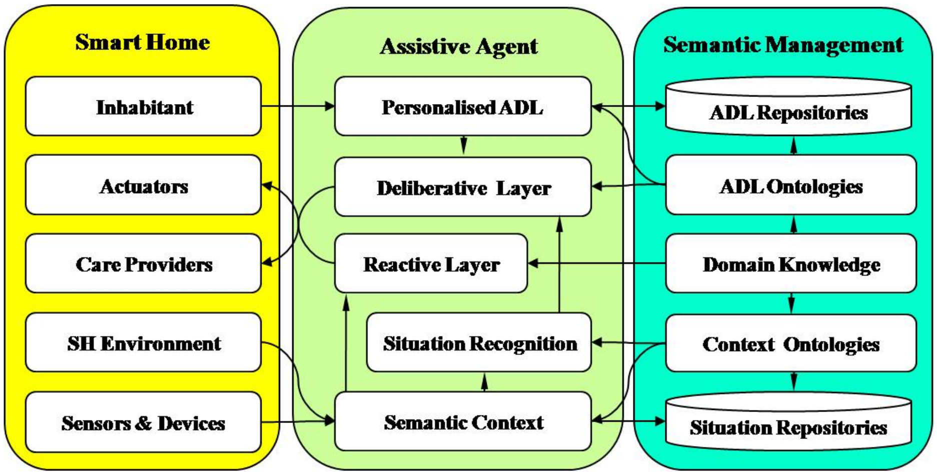


Fig. 1. The proposed system architecture

## 4 A Systematic Approach

Suppose that an actor performs activities and there is a sequence of state changes along the timeline as shown in Figure 2. In terms of the conceptualization of situation in the previous section, a situation at a specific time point $\tau$ can be described as the accumulation of states occurred before that particular time. This can be denoted as follows.

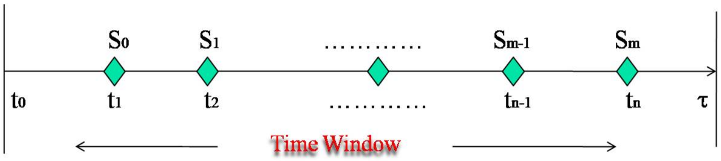


Fig. 2. The graphical representation of state traces

$$SITU_\tau \equiv S_0 \cup S_1 \cup ....... \cup S_{m-1} \cup S_m$$

Here $S_n$ denotes a state at the time point $t_n$, $SITU_\tau$ the situation at time $\tau$. In this way, the interpretation of a situation is essentially the joint interpretation of individual states. As each state is detected by a sensor, a state change is equivalent to a sensor's activation. If sensors can be semantically described, i.e., to give each sensor reading explicit meaning, it will be straightforward to generate situations with explicit semantics that can be interpreted by both humans and software agents

We contend ontological SH modeling lends itself naturally for semantic situation formation. The main reasons are three folds. Firstly, ontologies can provide rich descriptions for sensors, environments and activities. These attributes can disclose the inherent implicit knowledge, useful for situation construction and interpretation. For example, suppose a sensor is attached to a milk bottle within a freezer in the kitchen. When the sensor activation is detected, it is easy to infer that an actor is in the kitchen opening the freezer and take the milk bottle. In addition, semantic descriptions are understandable and processable for both humans and machines, thus supporting automated situation comprehension and inference. Secondly, ontologies can capture and model rich interrelationships between sensors, situations and activities. The interlinking facilitates semantically enabled data integration and fusion because situation awareness of complex dynamic environments like SHs often require to fuse information from multiple, disparate information sources for the recognition of a situation [19]. Thirdly, the embedded knowledge such as activity patterns, heuristics and causal relations in ontologies allow assistive systems to reason over perceived situational data with respect to the prediction of future states of SHs or next action of the inhabitant. Figure 3 depicts the core elements and technologies on semantic SH modelling, content creation and manipulation in the Semantic Management component in Figure 1. Details are described below.

### *4.1 Smart Home Analysis*

A SH is a complex ecosystem typically consisting of a physical environment with various furniture, household appliance, rooms, inhabitants that perform various ADLs within the environment, and sensors and devices (actuators) to sense and act on environmental changes and inhabitant behaviors. At any specific time it will generate data/information about the environment such as temperature, humidity, the status of doors, windows and lights, about the behaviors of inhabitants such as sleeping, cooking or watching TV and about events

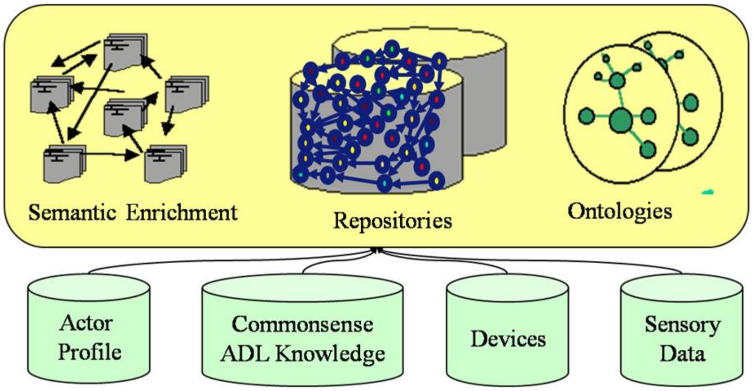


Fig. 3. The core components for semantic situation modelling

within the smart home such as alarm-fired, cooker-turn-on or tap-turn-on. Such information once monitored and collected can be aggregated to denote a situation against which an assistive system should be able to carry out interpretation and reasoning to make just-in-time assistances for the inhabitant. As such, the central issue is how to fuse data from multiple data sources to form a meaningful situation and further interpret them at higher level of automation, i.e., by software agents.

The nature of SH presents a number of challenges to situation formation and comprehension. Firstly, most sensor data are primitive numerical data such as 3-D coordinates for motion detectors, 2-state values for contact sensors. They lack formal descriptions and can only be consumed by humans through hard-coded operation logics in ad hoc data processing components. For example, for a contact sensor attached to a tap, the two state values, either on and off or 0 and 1, may denote different actions. A human user may be able to interpret that the state on/1 corresponds to the tap turn-on action, and the state off/0 the tap turn-off action. But the primitive signals or data will tell anything about this. Metadata is needed. Otherwise it is difficult, if not impossible, for machines or soft-agents to interpret and reason their high-level situational meaning. Secondly, sensor data are increasing available in a variety of diverse forms, such as unstructured textual data, audio and surveillance videos. They are heterogeneous in data formats and representation, and conceptually isolated from each other. For example, a location sensor (or a video monitor) can detect an inhabitant in front of cooker. An event sensor detects the turn-on of a cooker and a contact sensor detects the move of a spaghetti pack. While each sensory data reflect one facet of the situation, it requires the interlinking and fusion of data from multiple, disparate information sources in order to comprehend and understand such a complex situation.

In addition to situation construction and interpretation, the third challenge is how to model and represent normal ADL routines and inhabitants' profiles. Formal modeling and representation of ADLs and user profiles in essence provide a recognition context for an assistive system to interpret perceived situational data for the provision of personalized assistance. Traditionally activities are modeled as processes using probabilistic or statistical analysis methods, such as Markov Models and Bayes Networks. To construct a specific activity model for a specific individual, a large dataset obtained from monitoring the particular use's activities is required in order to train and test the model. This is usually done using machine learning techniques. However, this approach suffers from the problems of reusability and scalability, i.e., one model for one user is not applicable to another one; and every activity needs to be learned. Not mentioning the lengthy computation and accuracy issues, in most cases data are simply not available.

### *4.2 Context Modeling*

SH inhabitants perform ADLs in a diversity of temporal, spatial, environmental and event contexts within a SH. Spatial contexts consist of location information and surrounding entities such as rooms, windows, household furniture and appliance. Events contexts contain background activities and dynamic state changes of appliance and devices. Example events include the previous or ongoing activities of an inhabitant, the state changes of doors, windows, lights, alarms, a cooker and taps. Environmental contexts are composed of environmental information such as temperature, humidity and general weather conditions. Temporal contexts indicate the time and/or duration. Apparently there are close couplings between ADLs and contexts. For example, a cooking ADL happens in the kitchen with a cooker turned on. A grooming ADL takes place in washing room in the morning. Lights turn on in the evening and windows (or air-conditioners) open when temperature is high.

We build seven ontologies for a SH. These include an ontology for the physical equipment such as sensors, actuators, medical devices and home electronic or electrical appliances; an ontology for actions and ADLs such as watching television and making drinks; an ontology for living spaces and environments such as the kitchen, sitting rooms; an ontology for actors such as inhabitants, care-providers; an ontology for medical information; an ontology for software components such as services and applications and an ontology for time in order to model temporal information. Each ontology is used to explicitly conceptualise a specific aspect and overall they provide a semantic data model for the construction of SH situations. Figure 4 shows some classes and properties of SH ontologies which have been developed using the Protege [20] It is worth noting that existing well-defined ontologies could be imported and reused directly, for example the time ontology [21].

### *4.3 Situational Data Creation*

Ontologies are knowledge models that can be used to create semantic data. There are two major approaches for this purpose. One is to use generic ontology editing tools such as the Protege OWL Plugin [20]. These tools can perform several activities in one go, such as knowledge acquisition, ontology editing, knowledge population as well as knowledge base creation. They are feature rich but require professional knowledge engineering expertise. So this method is suitable for knowledge engineers. Another approach is to develop domain specific dedicated lightweight annotation tools for domain experts or resource (data) providers to carry out semantic annotation and create knowledge repositories. Such tools are often designed to provide intelligent semi(automatic) support for knowledge acquisition and modelling, including automated information extraction, classification and completion, to help create instances.

Given the nature of data in SH we develop a two phase semi-automatic approach to semantic descriptions. In the first phase data sources such as sensors and devices are manually semantically described. As the number of data sources in a SH is relatively limited, though

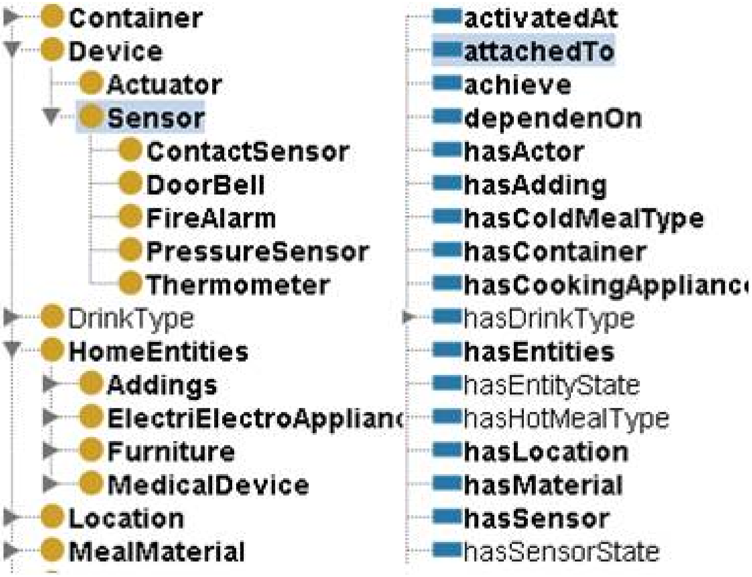


Fig. 4. A fragment of the SH ontology

large, it is manageable to create all semantic instances manually by generic ontology editors such as the Protg OWL Plugin. In the second phase dynamically collected sensory data are first converted to textual descriptors. For example, a contact sensor returns a two-state binary value. It can be pre-processed to literals sensible for denoting two states such as on/off or open/close or used/unused, etc. The concrete interpretation of the state depends on the purpose of the object to which the sensor is attached. For example, the two states of a contact sensor in a microwave could be open/close. If the contact sensor is attached to a milk bottle, the literal might be used or unused. The conversion of numerical values to descriptive terms is to facilitate interpretation and comprehension for both humans and machines. Pre-processed data can then be automatically attached to semantic instances of the corresponding data source to create a data repository.

### *4.4 Situational Data Storage*

Once semantic data are generated, they can be archived in semantic repositories for situation construction and interpretation. Semantic repositories are essentially knowledge bases consisting of millions of RDF triples. They are built on top of traditional database management systems by adding a semantic processing layer for semantic manipulation. Semantic repositories have been extensively studied and open source systems are available for use [22, 23].

Based on the nature of SH data we design a centralised repository with two interlinked components, as shown in Figure 5. The first component contains semantic descriptions relating to the various devices, inhabitants, individual SH and the services offered within an institution. These entities and their semantic descriptions are relatively stable for a care institution, i.e. static data. This component can functionally serve as a registry so that new SH once built within the institution, devices once added to any individual SH, inhabitants once they take residence in a SH and new services once developed can all be registered for later discovery and reuse. The second component is dedicated to the storage of dynamically generated sensory data and derived high level ADL data, which are time dependent, varying and extensive, i.e. dynamic data. Static data only need to be described and recorded once while dynamic data have the requirement to be recorded whenever they are generated. The separation of their storage saves storage space and also increases recording efficiency. Another advantage with this design is its ability to supports dynamic, automatic discovery of devices, device data, services and inhabitants, thus facilitating reuse of data and services. Further details of these concepts will be presented in the following Section.

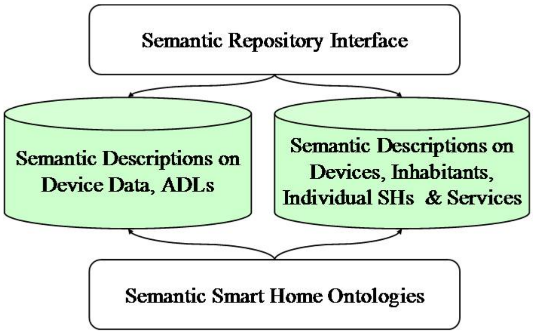

Semantic Repository Interface
Semantic Descriptions on Device Data, ADLs
Semantic Descriptions on Devices, Inhabitants, Individual SHs & Services
Semantic Smart Home Ontologies

Fig. 5. The semantic data repository

## 5 A Situation Aware Assistive Agent

As semantic data are machine understandable and processable, the assistive system is able to use an intelligent agent to automatically interpret situational data for activity recognition. The Assistive Agent, as shown in Figure 1, is responsible for the interpretation of the significance of perceived data and the provision of decision support for just-in-time ADL assistance. It performs reasoning against domain knowledge and subsequently advises corresponding actions to inhabitants. In the context of situation-aware assistive living, domain knowledge such as context, ADL and user profiles is formalized as Description Logic (DL) [24] based formulae in the form of subject-predicate-object triples, e.g. the event "FireAlarm" leads to "leadTo" the action "Call999". They can be described in ontological relationships and represented in RDF or OWL. The perception of an event and/or the detection of sensor signals are equivalent to the identification of a concrete instance of a class. For example, the activation of a contact sensor in a cup means that the cup, as an instance of Container, is used in an ADL. Suppose the Container class is the range of the hasContainer property, it can be inferred that the hasContainer property is assigned the value cup. If the hasContainer property is used to describe the MakeDrink class, it can be further inferred that a MakeDrink ADL has taken place. In this way the sensing of an agent amounts to the retrieval of the situational data periodically from the semantic repositories.

Central to situation-aware ADL assistance is the comprehension and reasoning capabilities of the Assistive Agent. In terms of the nature of a SH's situations the Assistive Agent can be internally designed in a two layer framework - refer to Figure 1. The Reactive Layer is used to deal with emergency situations such as an alarm fires or a pre-defined action takes place such as taking medicine at a specific time. Such situations usually involve fewer sensor data but require quick responses. The Deliberative Layer is responsible for the recognition of complex non-emergency situations that involve multiple sensor inputs. For example, sensors attached to a milk bottle, a kettle and a cup have been activated within a short time interval, how to decide the situation and further to assist the inhabitant with the completion of the ADL being performed.

An Assistive Agent comprehends perceived situational data by interpreting the data against their ontological context, i.e. ontologies. For instance, a smoke sensor in a lounge can be semantically described using two property-value pairs - [hasConsequence, fire] and [hasLocation, lounge]. Whenever the sensor is activated, an agent can interpret the occurrence against the above semantic context in the ontologies and recognize the situation "a fire breaks in the lounge". With recognized situation the future states of a SH can then be predicted and ADL assistance is subsequently provided through reasoning and inference. For example, a fire event can be semantically described with three property-value pairs - [takeAction, toEvacuate], [takeAction, callFireEngine] and [hasEffect, homeEvacuated]. Whenever a fire event is detected, the agent can reason against the above knowledge to advise the inhabitant to evacuate the home and call fire engines. It can further deduce that the home is empty. Reasoning at the Reactive Layer can be directly realized via built-in entailment rules in DL based ontologies.

A single sensor input can sometimes decide a specific situation, in particular for those emergency situations as discussed above. Nevertheless, most situations may involve perception inputs from multiple sources. In this case, a situation requires joint formation and interpretation of multiple perceived sensor data. For example, if sensors attached to a milk bottle, a teabag and a cup have all be activated within a short period, by linking what have happened it is reasonable to assume a situation that involves cup, sugar, milk and tea. It is straightforward for humans to figure out that this is a situation in which "MakingTea" ADL takes place. However, for software agents to recognize the situation as humans do, it requires an explicit representation of these situations and reasoning mechanisms. The reasoning mechanism will combine all sensor inputs to derive the corresponding situation by interpreting the aggregated perceived data against the abstract knowledge representation.

As an ADL can be viewed as a sequence of situations along the temporal dimension, we can model situations through semantic ADL modeling, i.e., to build an ADL ontology as discussed in Section 4.1. The ADL ontology consists of an ADL hierarchy in which each node, also called as a class, denotes a type of ADL as shown in Figure 6. Each ADL class is described with a number of properties and sub-classes can inherit all properties from its parent class. A property is defined by specifying its domain and range. The domain refers to all classes that can be described by the property and the range refers to all classes whose instances can be assigned to the property. A property describes a class using either a literal or an instance of another class as its value, thus linking two classes. This essentially gives rise to a description based activity/situation model, i.e. an ADL/situation is described by various properties. The underlying idea is that if a number of properties can be identified and linked, then the corresponding situation and ADL can be inferred.

The agent monitors and collects perceived sensor inputs by periodically retrieving semantic situational data from semantic repositories. These situational data have already been enriched with ontological relationships, thus ready for reasoning. The agent performs reasoning at the Deliberative Layer to derive the situation and its corresponding ADL. The process is as follows: Sensor inputs are used to identify concrete items that have been involved in ADLs. These items should have already been specified as instances of classes in SH ontologies. In terms of the scope of a property's range, the property that takes the identified item as its value can be inferred. In terms of the scope of a property's domain the ADL(s) that can be described by the inferred properties can then be recognized. As properties can be inherited from super-classes (higher level abstract ADLs) to sub-classes (lower level specific ADLs), the lower a class is in the ADL class tree the more properties it has. This means that the more sensor data that are available, the more accurately ADLs can be recognized. Conceptually the process amounts to the gathering of multiple sensor data at a specific time to form a situation. The situation is interpreted to identify the corresponding ADL and further identify these items in order to complete the ongoing ADL.

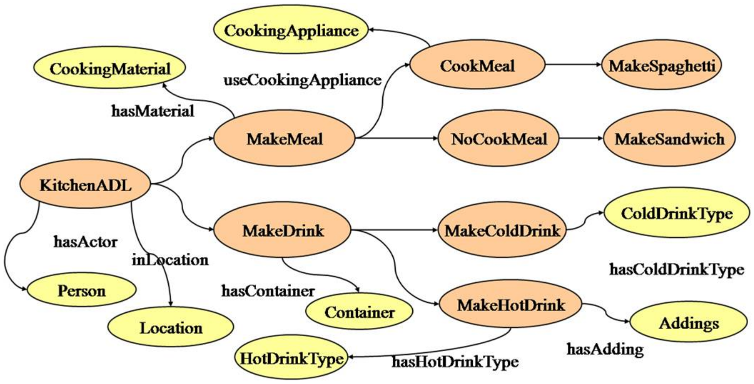
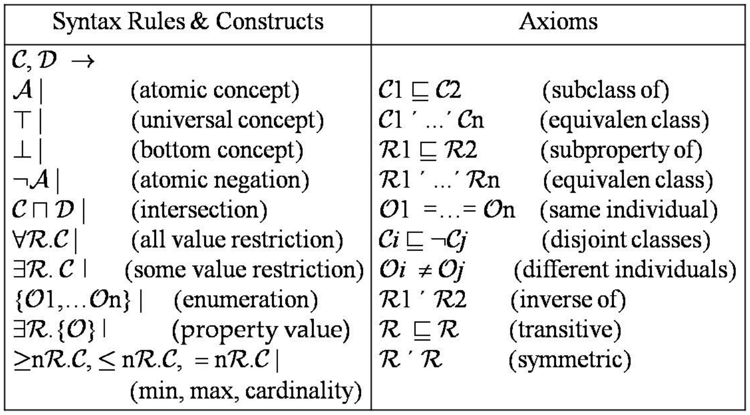


Fig. 6. A fragment of kitchen ADL hierarchy

The reasoning capabilities of the cognitive assistive agent are based on the theoretical foundation of Description Logic. Briefly, suppose that in abstract notation we use the letters $A$ for atomic concepts, the letter $R$ for atomic roles, the letter $T$ for TBox, and the letters $C$ and $D$ for concept descriptions. Concept descriptions in OWL can be formed using the syntax rules, constructors and axioms in Figure 7.

DL supports a number of reasoning tasks [25]. If we view a situation as a description for a unknown activity, then the interpretation of the situation is equivalent to the subsumption reasoning, i.e., to decide if a concept description $C$ is subsumed by a concept description $D$, denoted as $C \supset D$. The commonly used tableau algorithms [26] use negation to reduce subsumption to unsatisfiability of concept descriptions, which can be described below.

- Reduce subsumption to check unsatisfiability of concept description, i.e., C $\subseteq D \mapsto C \cap \neg D$
    - Check whether an instance b of this resulting concept description can be constructed
    - Build a tree-like model for the concept description



| Syntax Rules & Constructs | | Axioms | |
|---|---|---|---|
| $\mathcal{C}, \mathcal{D} \rightarrow$ | | | |
| $\mathcal{A} \mid$ | (atomic concept) | $\mathcal{C}1 \sqsubseteq \mathcal{C}2$ | (subclass of) |
| $\top \mid$ | (universal concept) | $\mathcal{C}1$ ′ ...′ $\mathcal{C}$n | (equivalen class) |
| $\bot \mid$ | (bottom concept) | $\mathcal{R}1 \sqsubseteq \mathcal{R}2$ | (subproperty of) |
| $\neg\mathcal{A} \mid$ | (atomic negation) | $\mathcal{R}1$ ′ ...′ $\mathcal{R}$n | (equivalen class) |
| $\mathcal{C} \sqcap \mathcal{D} \mid$ | (intersection) | $\mathcal{O}1$ =...= $\mathcal{O}$n | (same individual) |
| $\forall\mathcal{R}.\mathcal{C} \mid$ | (all value restriction) | $\mathcal{C}i \sqsubseteq \neg\mathcal{C}j$ | (disjoint classes) |
| $\exists\mathcal{R}.\ \mathcal{C} \mid$ | (some value restriction) | $\mathcal{O}i \neq \mathcal{O}j$ | (different individuals) |
| $\{\mathcal{O}1,\ldots\mathcal{O}\text{n}\} \mid$ | (enumeration) | $\mathcal{R}1$ ′ $\mathcal{R}2$ | (inverse of) |
| $\exists\mathcal{R}.\{\mathcal{O}\} \mid$ | (property value) | $\mathcal{R} \sqsubseteq \mathcal{R}$ | (transitive) |
| $\geq \text{n}\mathcal{R}.\mathcal{C}, \leq \text{n}\mathcal{R}.\mathcal{C}, = \text{n}\mathcal{R}.\mathcal{C} \mid$ | | $\mathcal{R}$ ′ $\mathcal{R}$ | (symmetric) |
| | (min, max, cardinality) | | |

Fig. 7. OWL Syntax Rules, Constructs and Axioms

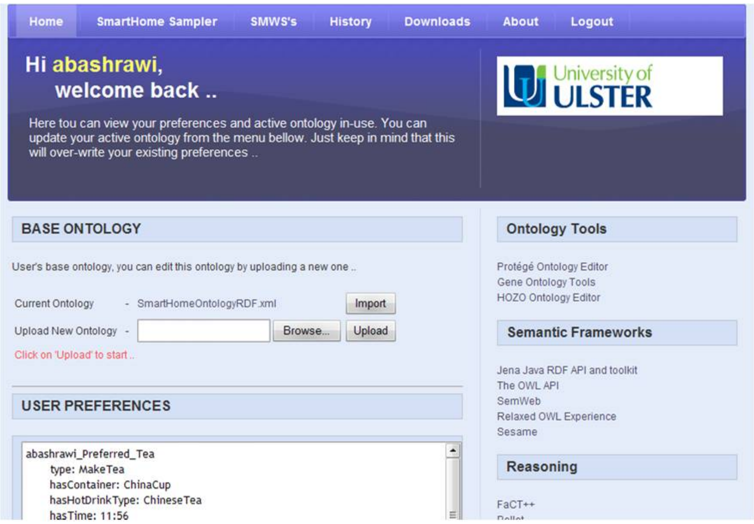
- Transform the concept description in Negation Normal Form
- Decompose the description using tableau transformation rules
- Stop when a clash occurs or no more rules are applicable
- If each branch in the tableau contains a clash, the concept is inconsistent

## 6 Implementation and Evaluation

We use the Kitchen ADL class hierarchy in Figure 6 to delineate how our approach works. As can be seen, KitchenADL is the top class of kitchen ADL with two properties - inLocation and HasActor. It has two subclasses, MakeDrink and MakeMeal. Apart from inherited properties, MakeDrink has a property of the class Container that could be a cup, a mug or a bowl. Similarly MakeDrink has two subclasses, MakeHotDrink and MakeColdDrink and each with some more properties. For example, MakeHotDrink ADL has two properties of the class HotDrinkType and Addings respectively. The HotDrinkType can assume one of tea, coffee or chocolate and the Addings can assume sugar and milk. Situation recognition that is denoted as corresponding ADLs is performed as follows:

Suppose that the contact sensor in a cup is activated. This means that the cup, as an instance of Container, is used in an ADL. As the Container class is the range of the hasContainer property, it can be inferred that the hasContainer property is assigned the value cup. Since the hasContainer property is used to describe the MakeDrink class, it can be further inferred that a MakeDrink ADL has taken place. Nevetheless, it is not possible to ascertain whether the ADL is MakeHotDrink or MakeColdDrink as both ADLs have the hasContainer property. This is exactly one of the advantages of the description based ADL recognition because based on limited sensor information the system can still identify uncertain high level ADLs. In the given example, though we can not tell the concrete ADL, i.e. the MakeHotDrink or the MakeColdDrink, we can at least know that the inhabitant is performing a MakeDrink ADL. Suppose we obtain another sensor data from a coffee container, then we can determine that the inhabitant is making coffee but we still do not know if it is a white coffee or a black coffee. Hence the sensor data from a milk or sugar container can further help to recognize the details of the performed ADL. From what we have described above, it is apparent that the proposed approach can monitor the unfolding of an ADL and dynamically build situations based on the underlying semantic data models. This will enable the assistive agent to incrementally recognize the ultimate ADL, which may be considered as not previously possible. The reasoning can be performed automatically using a DL-based reasoner such as the Fact reasoner [25].

We have implemented the proposed approach to situation-aware ADL assistive living in a feature-rich prototype assistive system. Figure 8 shows the front-end interface of the system. The system is developed with C# language as the scripting language while the front-end is developed using ASP.NET with Ajax and Silverlight support for better user experience. We use the SemWeb semantic library for C# [27] to read and write RDF, manage RDF in persistent storage, query persistent storage via simple graph matching and SPARQL, and make SPARQL queries to remote endpoints. SemWeb provides built-in general-purpose inference, but we use an implementation of the Euler proof mechanism for reasoning [28]. Euler is an inference engine supporting logic based proofs. It is a backward-chaining reasoner enhanced with Euler path detection.



Hi abashrawi,
welcome back ..

Here tou can view your preferences and active ontology in-use. You can update your active ontology from the menu bellow. Just keep in mind that this will over-write your existing preferences ..

BASE ONTOLOGY

User's base ontology, you can edit this ontology by uploading a new one ..

Current Ontology - SmartHomeOntologyRDF.xml

Upload New Ontology -

Click on 'Upload' to start ..

USER PREFERENCES

```
abashrawi_Preferred_Tea
    type: MakeTea
    hasContainer: ChinaCup
    hasHotDrinkType: ChineseTea
    hasTime: 11:56
```

Ontology Tools

Protégé Ontology Editor
Gene Ontology Tools
HOZO Ontology Editor

Semantic Frameworks

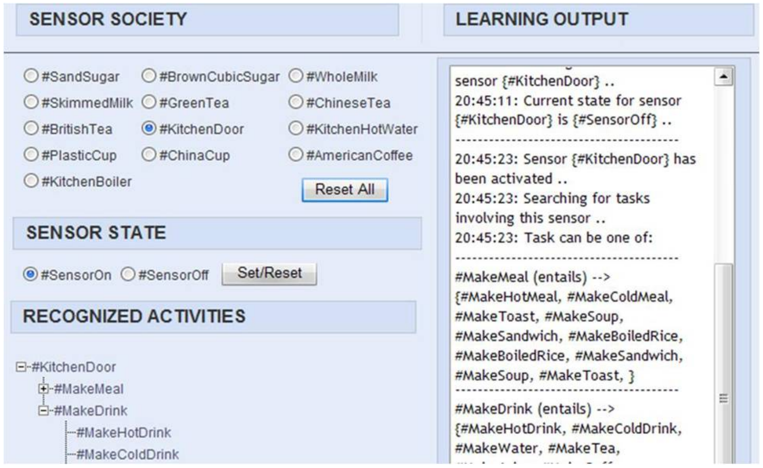
Jena Java RDF API and toolkit
The OWL API
SemWeb
Relaxed OWL Experience
Sesame

Reasoning

FaCT++

Fig. 8. The front-end interface of the Assistive system

The system works as follows: A user first logs into the system and uploads the SH ontologies from the BASE ONTOLOGY panel. By registration and logon the user establishes his/her identity. As such the user's ADL preferences can be browsed in the USER PREFERENCES panel as can be seen in Figure 7. Once SH ontologies are loaded, the system can display sensors that are semantically described. At this stage the system can operate in two modes - simulated and real-time ADL monitoring. In the simulated scenario, the system does not need to be connected to sensors. Sensor activation is simulated by the selection of a sensor, e.g. KichenDoor, in the SENSOR SOCIETY panel and the set-on of the sensor in the SENSOR STATE panel, see Figure 8. This is equivalent to the activation of real sensors. Once a sensor activation is observed, either simulated or triggered in real time, it will be used to form a situation to reason against the semantic ADL descriptions. The LEARNING OUTPUT panel displays the inference process of the assistive agent as sensors are activated and events perceived. The RECOGNISED ACTIVITIES panel displays the recognized ADL and its location in an ADL tree structure. Both are shown in Figure 9.



○ #SandSugar ○ #BrownCubicSugar ○ #WholeMilk
○ #SkimmedMilk ○ #GreenTea ○ #ChineseTea
○ #BritishTea ● #KitchenDoor ○ #KitchenHotWater
○ #PlasticCup ○ #ChinaCup ○ #AmericanCoffee
○ #KitchenBoiler

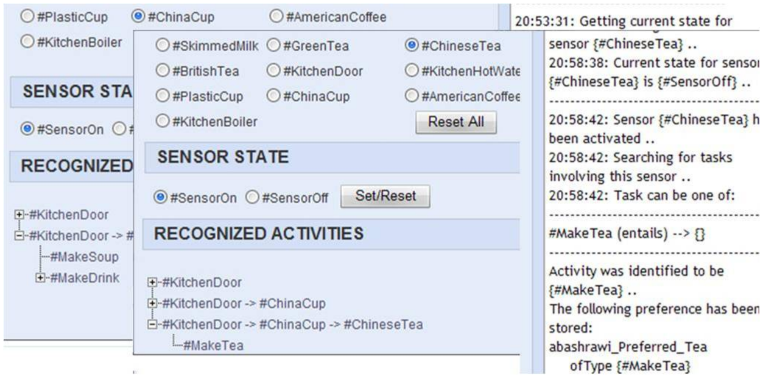
● #SensorOn ○ #SensorOff

```
⊟-#KitchenDoor
   ⊞-#MakeMeal
   ⊟-#MakeDrink
      ├-#MakeHotDrink
      ├-#MakeColdDrink
```

```
sensor {#KitchenDoor} ..
20:45:11: Current state for sensor
{#KitchenDoor} is {#SensorOff} ..
--------------------------------------
20:45:23: Sensor {#KitchenDoor} has
been activated ..
20:45:23: Searching for tasks
involving this sensor ..
20:45:23: Task can be one of:
--------------------------------------
#MakeMeal (entails) -->
{#MakeHotMeal, #MakeColdMeal,
#MakeToast, #MakeSoup,
#MakeSandwich, #MakeBoiledRice,
#MakeBoiledRice, #MakeSandwich,
#MakeSoup, #MakeToast, }
--------------------------------------
#MakeDrink (entails) -->
{#MakeHotDrink, #MakeColdDrink,
#MakeWater, #MakeTea,
```

Fig. 9. Simulated situation construction and ADL recognition

Figure 10 illustrates the dynamic situation formation and incremental ADL recognition process. When a KitchenDoor sensor is activated, only high-level ADL such as MakeMeal and MakeDrink can be inferred. When ChinaCup and ChineseTea sensors are activated later, situations with more contextual details can be dynamically formed. By reasoning these situations an assistive agent can recognise the ongoing ADL progressively in increasing details, e.g., MakeDrink initially and then MakeTea as depicted in Figure 10. Suppose that a user



○ #PlasticCup ● #ChinaCup ○ #AmericanCoffee
○ #KitchenBoiler

● #SensorOn

```
⊞-#KitchenDoor
⊟-#KitchenDoor -> #
   ├-#MakeSoup
   ⊞-#MakeDrink
```

○ #SkimmedMilk ○ #GreenTea ● #ChineseTea
○ #BritishTea ○ #KitchenDoor ○ #KitchenHotWate
○ #PlasticCup ○ #ChinaCup ○ #AmericanCoffee
○ #KitchenBoiler

● #SensorOn ○ #SensorOff

```
⊞-#KitchenDoor
⊞-#KitchenDoor -> #ChinaCup
⊟-#KitchenDoor -> #ChinaCup -> #ChineseTea
   └-#MakeTea
```

```
20:53:31: Getting current state for
sensor {#ChineseTea} ..
20:58:38: Current state for senso
{#ChineseTea} is {#SensorOff} ..
--------------------------------------
20:58:42: Sensor {#ChineseTea} h
been activated ..
20:58:42: Searching for tasks
involving this sensor ..
20:58:42: Task can be one of:
--------------------------------------
#MakeTea (entails) --> {}
--------------------------------------
Activity was identified to be
{#MakeTea} ..
The following preference has been
stored:
abashrawi_Preferred_Tea
   ofType {#MakeTea}
```

Fig. 10. The incremental situation formation and ADL recognition process

has a pre-defined, semantically described preferred ADL UserAMakeTea. By comparing the user's MakeTea profile with the perceived situation, an assistive agent can infer what shall be done next in order to complete the ongoing ADL, thus providing situation-aware personalized ADL assistance for the particular user. For example, if abashrawi-preferred-tea ADL contains sugar, the agent may remind the user to add sugar if it does not detect the activation of the sugar container for a pre-defined period of time.

On the other hand, if a user activity has been recognized repeatedly over a relatively long period of time, and there is no corresponding matching ADL profile, the activity can be recorded as a user's preferred ADL profile. This is the learning process. We shall not discuss it here in details due to space limits.

In addition to evaluate the approach and system in the simulated scenario, we have designed an experiment in our smart home environment for evaluation of the proposed approach and the implemented system in a real world use case. We attach contact sensors to teabag, sugar, kettle, milk and cup containers. Then we connect the prototype ADL assistive system to the sensors via the Tynetec wireless receiver. The experiment runs as a user performs making tea activity following the scenario discussed above, i.e., first coming to the kitchen, then taking a cup, etc. Each time the user takes an action/item, the sensor activation is perceived and passed to the assistive system. The system operates and produces results the same as we discussed in the simulated scenario.

We have designed a number of activity scenarios for testing and evaluation purpose. These include different activities, e.g., making tea and preparing pasta, and interweaved activities such as a phone call or fire alarm occurs during making tea. We also test features that are not covered here, e.g., learning user activity profiles, using different assistance prompts. Initial results are very positive. The system worked well with the diversity of scenarios and is able to recognize corresponding activities. This proves the approach and system are applicable in real world application scenarios.

Semantically enhanced situation-aware ADL assistance has a number of compelling advantages: Firstly, the scalability of situation modeling has been a bottleneck to effective situation aware applications. It is often the case that proof-of-concept experiments, either state-based or process-based approaches, work well but fail to scale up. The use of ontological ADL modeling as a way of situation modeling overcomes this problem. Ontology engineering offers extensive technological support, including tools, APIs, storage and reasoners. Ontologies of thousands of classes have been developed in other domains, e.g. 7,000 concepts in the gene ontology, and semantic data repository of 25 million triples has been practiced in TripleStore [23]. For smart homes, ADL classes and associated instances are simply not present in such a scale. Secondly, semantic ADL models contain explicit rich semantics and built-in logical entailment rules. This allows not only humans but also assistive software agents to interpret, comprehend and reason against semantic situational data. As such, situation monitoring and ADL recognition can be realized at higher levels of automation. Thirdly, description based reasoning provides a mechanism to dynamically construct situations by interpreting limited or incomplete sensor data that ultimately leads to the incremental recognition of the corresponding ADL. This capability is particularly important because assistive systems are supposed to provide reminding or suggestive assistances with limited sensory data.

## 7 Conclusions

In this paper we propose a semantic-enabled agent-based novel approach to enhanced situation-aware assistive living. We have discussed the concept of situation awareness and introduced an integrated system architecture for semantically enhanced situation awareness and intelligent just-in-time ADL assistance provision. We have analysed the nature and characteristics of SH-based assistive living. Based on the analysis we describe semantic situation modeling and formation including SH ontologies, semantic data creation and storage. We have presented the use of assistive agents for situation comprehension and ADL recognition with special emphases on the agent's internal structure and its interpretation and reasoning mechanisms. A simple yet convincing example scenario from a real world ADL assistance context has been used to illustrate our approach.

We have implemented a prototype assistive system for the proposed approach using the latest semantic technologies and toolkits. We have carried out both simulated and real world use case study. While the full evaluation of the proposed approach and system awaits further large-scale deployment and experimenting with real world users, initial research results have been promising. Our future work aims to address temporal issues such as parallel / concurrent ADL recognition. We shall extend the existing assistive system with capabilities of taking actions, e.g., playing audio/video or switch on/off devices/appliances through actuators.

## References

1. Weiser, M. (1991) "The computer for the twenty-first century", Scientific American, Vol. 265, No. 3. pp.94-104.
2. Boger, J., Poupart, P., Hoey, J., Boutilier, C., Mihailidis, A., 2005. A Decision-Theoretic Approach to Task Assistance for Persons with Dementia. In Proc. of the International Joint Conference on Artificial Intelligence, pp.1293-1299.
3. Bouchard, B., Giroux, S. 2006. A Smart Home Agent for Plan Recognition of Cognitively-impaired Patients, Journal of Computers, Vol.1, No.5, pp.53-62.
4. Jennings, N. R. and Wooldridge, M. J. 1998. Agent Technology: Foundations, Applications and Markets, Springer Verlag. ISBN 3-540-63591-2, pp325.
5. Dey, Anind K. 2001. Understanding and Using Context, Personal Ubiquitous Computing Vol.5, No.1, pp.4-7.
6. Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F.A. and Tanca, L. 2007. A data-oriented survey of context models, ACM SIGMOD Rec., Vol.36, pp.19-26.
7. Preuveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole. P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K. 2004. Towards an Extensible Context Ontology for Ambient Intelligence, Springer, LNCS vol. 3294, pp.148-159.
8. Chen, H., Finin, T., Joshi, A. 2003. An Ontology for Context-Aware Pervasive Computing Environments, Knowledge Engineering Review, Vol.18, no.3, pp197-207.
9. Gu, T., Pung, H.K. and Zhang. D.Q. 2005. A service-oriented middleware for building context-aware services, Journal of Network and Computer, Applications, Vol.28, No.1, pp.1-18.
10. Klein, M., Schmidt, A. and Lauer, R. 2007. Ontology-Centred Design of an Ambient Middleware for Assisted Living: The Case of SOPRANO, In proceeding of Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU), 30th Annual German Conference on Artificial Intelligence (KI 2007).
11. Latfi, F., Lefebvre, B. and Descheneaux, C. 2007. Ontology-Based Management of the Telehealth Smart Home, Dedicated to Elderly in Loss of Cognitive Autonomy, Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions.
12. The EU ASK-IT IP project, http://www.ask-it.org/

13. The EU SAPHIRE project http://www.srdc. metu.edu.tr/webpage/projects/saphire/ index.php
14. Sycara, K., Paulucci, M. and Lewis, M. 2003. Information discovery and fusion: semantics on the battlefield, Proc. of the 6th International Conference on Information Fusion, pp.538-544.
15. Smart, P.R., Bahrami, A., Braines, D., McRae-Spencer, D., Yuan, J. and Shadbolt, N.R. 2007. Semantic Technologies and Enhanced Situation Awareness, Proc. of the 1st Annual Conference of the International Technology Alliance.
16. Matheus, C. 2005. Using Ontology-based Rules for Situation Awareness and Information Fusion, Proc. of the W3C Workshop on Rule Languages for Interoperability. http://www.w3.org/2004/12/rules-ws/paper/74/.
17. Endsley, M.R. 1995. Toward a theory of situation awareness in dynamic systems, Human factors, Vol.37, pp.32-64.
18. Croft, D.G., Banbury, S., Butler, L.T. and Berry, D.C. 2004. The role of awareness in situation awareness, in A Cognitive Approach to Situation Awareness: Theory and Application, S. Banbury and S. Tremblay, Eds. pp.82-104. ISBN: 978-0-7546-4198-8.
19. Boury-Brisset, A.C. 2003. Ontology-based approach for information fusion, In Proc. of the 6th International Conference on Information Fusion, pp.523-529.
20. The Protg knowledge engineering framework, http://protege.stanford.edu/
21. The time ontology in OWL, http://www.w3.org/TR/owl-time/
22. Horrocks, I., Li, L., Turi D. and Bechhofer, S. 2004. The instance store: DL reasoning with large numbers of individuals, Proc. of the 2004 Description Logic Workshop, pp.31-40.
23. Harris, S. and Gibbins, N. 2003. 3store: Efficient Bulk RDF Storage, Proc. of the 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03), pp.1-15.
24. Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Schneider, P.P. 2003. The Description Logic Handbook - Theory, Implementation and Applications. Cambridge University Press.
25. Horrocks, I., Sattler, U. and Tobies, S. 1999. Practical reasoning for expressive description logics, LNAI, No.1705, H. Ganzinger, D. McAllester, and A. Voronkov, eds, pp.161-180.
26. Tsarkov, D., Horrocks, I.: FaCT++ Description Logic Reasoner: System Description, In Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006), LNAI4130, pp.292-297, 2006.
27. Semantic Web/RDF Liberary for C#.NET, http://razor.occams.info/code/semweb/
28. Euler proof mechanism, http://www.agfa.com/w3c/euler/