Building Semantic Robot Space based on the Semantic Web

Minsu Jang, Joo-Chan Sohn, Youngjo Cho
Electronics and Telecommunications Research Institute
Gajeong-dong 161, Yuseong-gu, Daejeon-si, 305-700, South Korea
Email: {minsu, jcsohn, youngjo}@etri.re.kr

***Abstract*—The robot space is a ubiquitous environment in which the networked robot plays the role of main mediator that accepts the sensor data from the environment, interprets the data to understand the situation, and decides the actions to perform. Semantic robot space provides the high-level data model of the robot space that enables context-aware service execution. We designed the semantic robot space based on the semantic web technology, which provided crucial advantages in architecting dynamic reconfiguration mechanism. The standard nature of the technology leverages the interaction among stake-holders in building robot spaces, which makes it possible to establish public registry-based construction of the semantic robot space. We conclude that semantic robot space based on the semantic web not only embellishes the intelligence of robots but also reduces the cost of installing and maintaining robot spaces.**

I. Introduction

Ubiquitous robotic space(URS) refers to the ubiquitous environments in which robots gain enhanced perception, recognition, decision, and execution capabilities through the interactions with networked and distributed sensors, actuators, and information services [1], [2]. Usually, a URS make up a very complex dynamic system which is composed of sensors, digital appliances, controllers, and web services as well as physical objects and spaces tagged by RFIDs or active beacons. Moreover, some of these URS participants are dynamically introduced into or removed from the URS.

A robot in the URS should interact well with the URS participants to properly perform its tasks e.g. navigating between regions, controlling the devices, or providing services to its users. It has been recognized that it's important to build an infrastructure to guarantee communication with compatible data among ubiquitous space participants that are built on heterogeneous technologies. Some have built dedicated middlewares [3], utilized web services [4] or standard protocols like Jini [5] and uPnP [6].

Based on the middle-ware, ubiquitous participants can exchange data. But, how do they know the meaning of the data? For example, if a temperature sensor sent a float number to an intelligent agent, how does the agent know the number represents the temperature? In the usual case, the agent should have been pre-programmed with a dedicated logic, *e.g. atmospheric mood regulation*, to properly process the number. But what if a new humidity sensor is deployed? The logic itself does not know the meaning of the numbers sent from the humidity sensor and the agent fails to process the number properly. As such, a feature called *dynamic reconfiguration* is an important issue for ubiquitous environments which is inherently dynamic. For example, it would be better for the agent to retrieve the identity of the data sender to determine the meaning of the number. Or even better, the agent would retrieve the identity and the context, *e.g. the position of the sender*, to determine not only the meaning of the number but also the composite meaning of the data from different sensors in the same region. A number of ubiquitous computing literatures describe semantic frameworks to achieve this kind of situation-aware operations. The core of the frameworks is to build up a semantic layer upon the ubiquitous environment [7], [8], [9].

We conjecture that the robot is the candidate for a central mediator in the ubiquitous environment, and the semantic layer will benefit the operations of the robot by the rationale . Robots will play the role of the main interface to the ubiquitous space through which human users interact with the ubiquitous space. That is because robots are devised with multitude of human-oriented interfaces like voice, gestures, simple remote controllers etc. Also, it should be noted that robots are (made to be) cognitive and social. As the main interface to the ubiquitous space, robots need to keep track of the configuration of the ubiquitous space which dynamically changes. Robots consult the sensors for context data (or be notified by the sensors), interpret the data to understand various situations around the space, determine context-aware actions to perform, and execute the actions by itself or via other ubiquitous participants. As such, robots need to keep track of the configuration of the ubiquitous space, know the meaning of the data generated from the space, and intelligently decide a series of actions to perform.

We have designed a framework called *semantic robot space (SRS)* and built some part of it. We describe in this paper the framework and benefits it brings to the ubiquitous robot space. SRS provides robots with capabilities to dynamically build the semantic configuration of the ubiquitous robotic space, restructure the configuration by dynamically discovering and deploying context-aware services, and interpret the context and perform actions in a situation-aware fashion. SRS is built based on the semantic web technologies, which is on par with the approaches given by [7], [8], [9].

By committing to the standard languages of the semantic web, it's possible to promote data interoperability among

978-1-4244-1635-6/07/$25.00

stake-holders around robot space implementations *e.g. robot and ubiquitous device manufacturers, service providers, construction companies etc.* The interoperability allows more flexible and interactive models of building semantic robot space as well as semantically rich models of service provision. We show how the semantic web technologies are applied and what benefits they bring.

Figure 1 summarizes our approach of applying semantic web standards to promote interoperation among various stakeholders in service provision via ubiquitous robots.

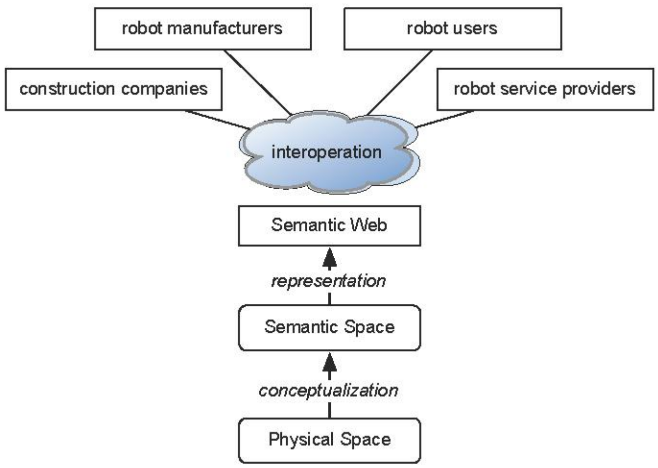

Fig. 1. Semantic robot space is a conceptual representation of the physical robot space, which is specified by the semantic web standards, which in turn promotes interoperation among stake-holders in service provision via ubiquitous robots.

The rest of this paper is organized as follows. We provide the purpose and the architecture of the semantic robot space in section II. We show how context-aware service execution is organized in section III, and explain how robot space is built and restructured dynamically in section IV and V, and then conclude in section VI.

II. Semantic Robot Space (SRS): Overview

Robot space, in this paper, is divided into two layers: *physical robot space* and *semantic robot space*. Physical robot space represents the layer of physical entities, quantified descriptions of the physical entities, and functionality implementations for accessing and controlling the physical entities. SRS is defined as an information overlay upon physical robot space, which conceptualizes the physical entities and functionalities. Operations in SRS include, among others, context interpretation, action planning, and user interaction, all of which built upon the symbolic knowledge representation and reasoning. Figure 2 shows an example of the layered structure by showing how the space is recognized differently in each layer.

The symbol, `sensor001`, in the SRS, is accompanied by a sensor profile that states the identities and the capabilities of the sensor, which is similar to the following sentence. The `pid` in this profile identifies the ID of the sensor in the physical robot space.

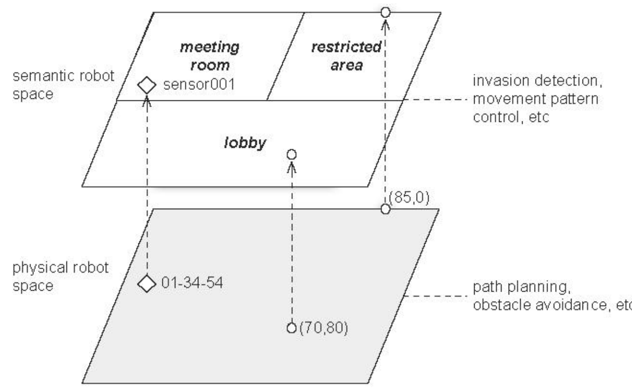

Fig. 2. Coordinates in the physical robot space is modeled as symbolic tags in SRS e.g. the coordinate `(70,80)` is interpreted into `lobby`. Services in the semantic layer include intrusion detection and movement pattern control which are performed according to the type of the region. Operations in the physical layer include path planning and obstacle avoidance which are performed based on the physical configurations.

```
sensor001(type=Sensor,
   output=Temperature,
   pid="01-34-54")
```

Any service built upon SRS, when it receives a data from the sensor with an ID `01-34-54` can determine the meaning of the data by referring to the sensor profile shown above. Also, if the sensor is a newly installed device, SRS can initiate a dynamic reconfiguration procedure to revise the service repertoire of the SRS. As a result, a new service may be deployed in the SRS to properly process the data from the new sensor. The profiles serve the knowledge cue for adapting service configurations upon dynamic changes in SRS.

Figure 3 shows the conceptual architecture of SRS. SRS is composed of two function bundles: *context-aware service execution (CASE)* and *dynamic reconfiguration (DR)*. CASE accepts sensor data from the physical robot space, interprets the data to understand situations, and feed a set of control commands back to the physical robot space if necessary. DR accepts the events notifying some changes of the configuration in the physical robot space, and adjusts the structure of the context interpretation network or robot service knowledge according to the changes. Basically, DR tries to dynamically deploy new services into or remove services from CASE. If a change in the physical robot space involves a new device installation, DR retrieves a profile for the device from the *device repository*, and then consult the *service repository* to discover any services that can perform on the new device. The services thus discovered are candidates for deployment.

III. Context-Aware Service Execution

We first give descriptions on context-aware service execution(CASE) of SRS. CASE performs situation-aware actions based on the semantic specifications, e.g. semantic space map and robot service knowledge, on the situation and the services.

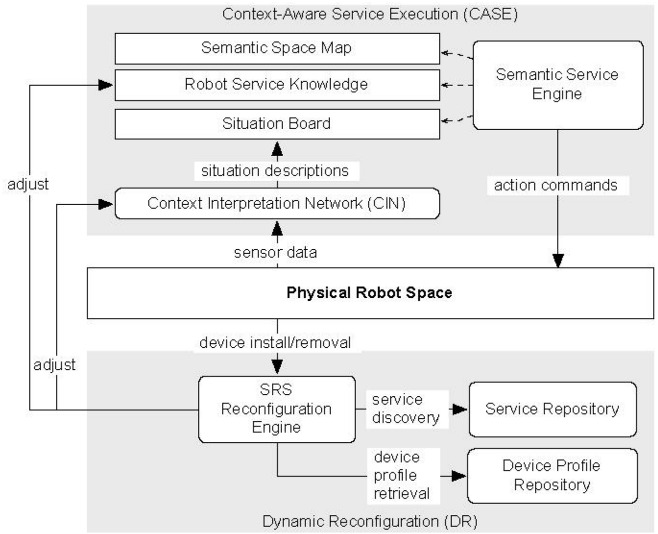

Fig. 3. SRS is composed two main function bundles: *context-aware service execution* and *dynamic reconfiguration*.

A. Semantic Space Map

The semantic space map is a semantic description of the spatial configuration of the robot space. Figure 2 presented a simple example of how physical region is mapped to semantic data - symbols - of SRS. In SRS, the symbols to describe spatial regions are defined as an ontology, which is specified in W3C Web Ontology Language(OWL) [10]. In the semantic space map, each region in the robot space is specified by a unique symbol and a set of semantic descriptors. The following is a sample description to describe a meeting room.[1]

```
region001 rdf:type  MeetingRoom;
          rdf:type  RestrictedArea;
          s:x       "0"^^xsd:integer;
          s:y       "45"^^xsd:integer;
          s:width   "30"^^xsd:integer;
          s:height  "30"^^xsd:integer.
```

The description states that `region001` is a type of meeting room and it is a restricted area. Also, it specifies the regional coordinates that are used to map physical regions into semantic region symbols.

Semantic map is used to interpret the situations occurring in specific regions or to take actions according to the relevant region type. For example, a CASE implementation can slow down or speed up the movement according to the type of the region a robot is moving across, or it can alarm the users if an intrusion is detected in one of the restricted area.

B. Robot Service Knowledge

The robot service knowledge contains context models and service rules. Context models contain factual descriptions relevant to the service, e.g. employee records, patrol routes for days of week etc for company surveillance service. Service rules specify which actions to take on which situations. In the rules, situations are tested at the condition part and the actions are initiated at the conclusion part. Internally, each condition of the rule is matched to the situational facts contained in the semantic space map and the situation board. The following is a simple rule for intrusion detection.[2]

```
rule intrusion-detection is
if
  EntranceLocation(?ent,?location)
  and RestrictedArea(?location)
  and
    (
     not actor(?ent,?act) or
     (actor(?ent,?act)
      and not admitted(?location,?act))
    )
then
  Intrusion(?ent,?location);
```

The rule fires an intrusion event if an entrance is detected at a location which is a restricted area, and the actor of the entrance event is not identified or the actor is not admitted to the location. `RestrictedArea` comes from the semantic space map; `EntranceLocation` and `actor` dynamically come from context interpretation; and `admitted` comes from robot service knowledge.

C. Context Interpretation Network (CIN)

The context interpretation network is constituted with a set of rules that conceptualize and synthesize sensor data gathered from the physical robot space. Conceptually, CIN is organized as a hierarchically layered context interpreters. The nodes at the lowest layer receives raw sensor data and the nodes at the highest layer generate the situational facts. Figure 4 shows a simple example of a CIN that generates facts about atmospheric mood from temperature and humidity.

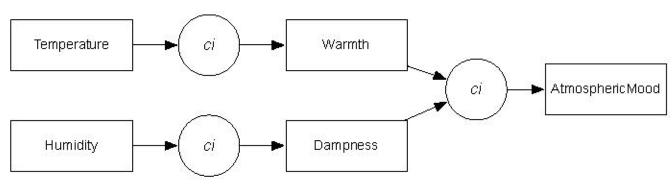

Fig. 4. A sample CIN for deriving atmospheric mood from temperature and humidity.

The nodes, denoted by *ci*, perform context interpretation, by accepting sensor data or lower contexts and generating higher contexts. Lower contexts are generated at the lower layers in the CIN, and higher contexts at the higher layers. The situational facts generated from all the layers of the CIN

[1]The sentences are written in an OWL syntax called N3 [11]. `s:` and `rdf:` are namespace prefixes to localize the identifiers. For details, please refer to [11].

[2]Identifiers with a question mark, e.g. `?ent`, denote variables.

are stored into the situation board to be used for context-aware action decision.

D. Semantic Service Engine

The semantic service engine performs reasoning over all the situational descriptions and the service knowledge. The core of the semantic service engine is the reasoning capability. All the semantic data of CASE is specified in OWL and rules, the reasoning capability should include ontology reasoning as well as rule processing. If we view the semantic data models as a composite context-aware service application, the semantic service engine can be regarded as an interpreter that executes the application. The output of the execution is a series of action commands that can initiate operations on robots and devices in the physical robot space.

IV. Dynamic Reconfiguration

As explained in section I, robots need to keep track of the changes of the configurations of the physical robot space to properly process all the sensor data which results in better understanding of the situation, and to adjust the action commands that can have valid effects. Dynamic reconfiguration is the mechanism to adjust the configuration of the service knowledge and the context interpretation network of the CASE. The adjustments change the way SRS responds to the situations.

Figure 5 shows the basic flow of operation in dynamic reconfiguration.

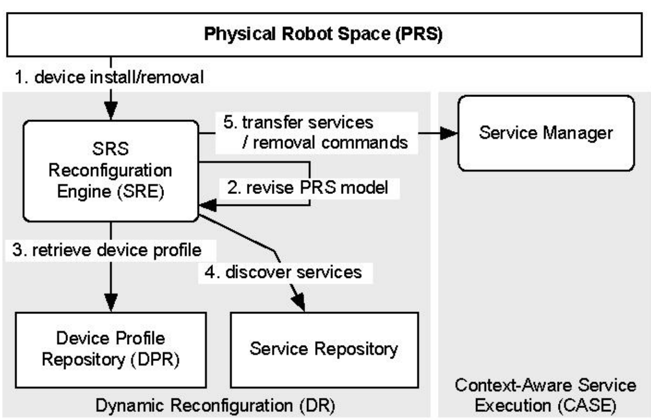

Fig. 5. A schematic flow of operation in dynamic reconfiguration.

The SRS Reconfiguration Engine (SRE) collects all the configurational events from the physical robot space and maintains the data model that describes the configuration. The model, called physical robot space model(PRS model), contains the information on the devices installed in the physical robot space.

The basic operation of DR is to get the events on the changes of the PRS; reflects the changes into the PRS model, by retrieving the device profiles via the device profile repository(DPR) if necessary; discover new services that can be deployed into the SRS; and transfer the services unto the service manager of CASE. The service manager properly deploys the services transferred.

An important issue in dynamic reconfiguration is the mechanism of service discovery. We adopted *means-end planning* as the basis of the service discovery, which is the most widely used planning mechanism. In means-end planning, operations should first be defined and described. Each operation is described by specifying the preconditions and inputs that are necessary for the operation and the goals and outputs that the operation can achieve. By matching the inputs of an operation and the outputs of another operation, operations can be composed into a goal-directed sequence of operations, a plan.

Based on the schema of the means-end planning, we modeled the robot service knowledge and the context interpreters as operations. Also, sensor devices are modeled as the operations that produce outputs. For example, a temperature sensor is modeled as an operation that outputs temperature values; and an intrusion detection service is modeled as an operation that accepts entrance notification, semantic spatial map, and security policy, and produce an intrusion notification.

Input-output matching is the key mechanism of service discovery in DR. A distinct point in service discovery of DR is that it discovers the services that *can be deployed*, which is contrary to the usual case of discovering the services that can satisfy a specific goal among the services that are already deployed and announced. Figure 6 explains the basic model of operation description and the concept of service discovery in DR with a simple example.

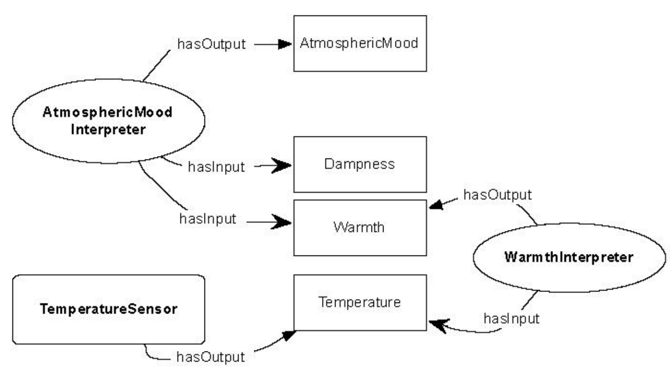

Fig. 6. Inputs and outputs of each operation serve as cues for producing a series of interconnected operations. In this model, *TemperatureSensor* produce temperature that is required by *WarmthInterpreter*. When a temperature sensor is installed, *WarmthInterpreter* becomes a candidate that can be deployed unto the CASE.

The input and output descriptions for sensors and devices are called *device profiles*, and those for services, including context interpreters, are called *service profiles*. As mentioned earlier, in SRS, all of these profiles are specified in OWL, which are served by the corresponding repositories - device profile repository(DPR) and service repository(SR). OWL is the W3C recommendation that are used to author semantic data on the web, so adopting the language potentially establishes a globally interoperable infrastructure for data exchange. We explain how this benefits in building SRS in the next section.

V. Building Semantic Robot Space Interactively

An important success factor for this kind of robot-based ubiquitous service is to reduce the cost of installing and maintaining the system. In SRS, we pursued the goal of reducing the cost by designing the automated SRS building via the DR mechanism explained in section IV. Based on the DR mechanism, installation is a simple two-step process of attaching or arranging the sensors, devices, and robots in the robot space, and turning on all of them. As the robot turns on, SRS implemented in the robot starts to detect the identifiers of the sensors and devices deployed in the space. Detecting new devices, it initiates the dynamic reconfiguration process of fig. 5, which results in setting up a service repertoire (*CIN and robot service knowledge*) in the CASE (*refer to fig. 3*).

The sources of device profiles and service profiles are important for the automated installation/maintenance scenario. It should be feasible to retrieve the profiles for new kinds of sensors or devices and the services supported by the them. So the device repository and the service repository should be open services to promote profile publishing is encouraged. This applies the same to the semantic space map. Building the semantic space map as well as physical space map is very costly in general. The space maps can be published and consumed in the same way as the profiles. Utilizing public registry/repository for profiles is becoming common in the device independence and RFID community, and we argue that the approach benefits ubiquitous or networked robot services as well.

Again, an important factor to establish publicly accessible profile, service, and map registry/repository is the *standard data format*, as *interoperability* is the foremost requirement in exchanging data. The use of the standard language of the semantic web, OWL, in modeling all the semantic data models of the semantic robot space gives advantages in this regard. Figure 7 shows how different stake-holders interact in building semantic robot spaces. As shown in the figure, construction companies publish default semantic (as well as physical) space maps for the buildings they build, which in turn can be amended by the users to reflect the changes they made to the space. The maps are imported and merged into the semantic data model of the SRS for use in its situation-aware operations. Device profiles authored by the manufacturers and published to the public, which in turn can be retrieved by the SRS for dynamic reconfiguration. Services and service profiles are authored by the service providers. The services are open to service discovery by any SRS.

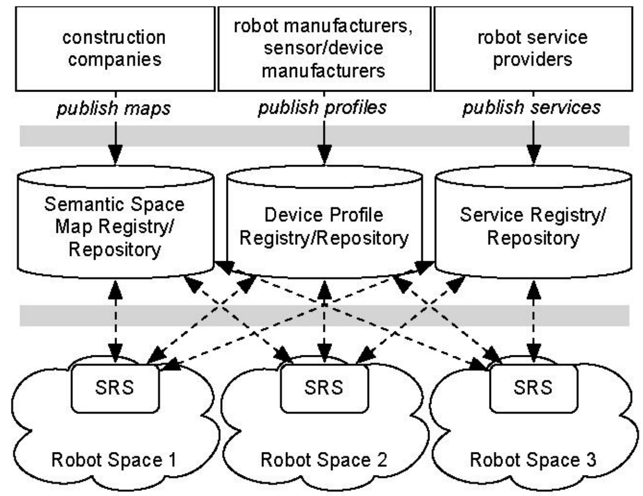

Fig. 7. The use of standard semantic web language for specifying profiles and maps enables data interoperability among different stake-holders relevant in building robot spaces, which in turn promotes the feasibility of the dynamic reconfiguration of the SRS. The interoperability layer is indicated by the grey bars.

VI. Conclusion

We introduced the concept of semantic robot space. Semantic robot space is composed of two main features: context-aware service execution and dynamic reconfiguration. To enable dynamic adaptation of the context-aware services upon installation or removal of devices, semantic data models are necessary. Semantic data models in the semantic robot space include device profiles, service profiles, and semantic space map. These semantic data enable automated installation and maintenance of the robot space, which reduces the cost of maintaining robot spaces. We employed semantic web language, OWL, to model and process the semantic data. The use of OWL makes it possible to efficiently exchange the semantic data among diverse stake-holders in building robot spaces. As a whole, we conclude that our design of the semantic robot space not only leverages the intelligence of robots but also the feasibility of the robot spaces as a real-world service model.

VII. Acknowledgement

This work was supported by the IT R/D program of MIC/IITA. [2007-S001-01, USN-based Ubiquitous Robotic Space Technology Development].

References

[1] Y.-G. Ha, J.-C. Sohn, Y.-J. Cho, and H. Yoon, "Towards ubiquitous robotic companion: Design and implementation of ubiquitous robotic service framework," in *ETRI Journal*, pp. 666–676, December 2005.

[2] M. Jang, J.-C. Sohn, and Y. Cho, "A semantic framework of smart environment comprehension for the ubiquitous robotic space," in *Proceedings of the 2006 International Symposium on Flexible Automation*, July 2006.

[3] A. Kofod-Petersen and M. Mikalsen, "An architecture supporting implementation of context-aware services," in *Proceedings of the Workshop on Context Awareness for Proactive Systems*, (Helsinki, Finland), June 2005.

[4] G. Austaller, J. Kangasharju, , and M. Muhlhauser, "Using web services to build context-aware applications in ubiquitous computing," in *In Piero Fraternali, Nora Koch, and Martin Wirsing, editors, Web Engineering 4th International Conference*, 2004.

[5] P. Huang, V. Lenders, P. Minnig, and M. Widmer, "Jini for ubiquitous devices."

[6] G. Schneider, C. Hoymann, and S. Goose, "Adhoc personal ubiquitous multimedia services via UPNP," in *Proceedings of the IEEE International Conference on Multimedia and Expo, 2001*, Aug 2001.
[7] X. Wang, J. S. Dong, C. Chin, S. R. Hettiarachchi, and D. Zhang, "Semantic Space: An infrastructure for smart spaces," in *Pervasive Computing*, pp. 32–39, IEEE, Jul-Sept 2004.
[8] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," in *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 2003.
[9] R. Krummenacher and T. Strang, "Ubiquitous semantic spaces," in *Conference Supplement of the 7th Int'l Conf. on Ubiquitous Computing*, Sept 2005.
[10] D. L. Mcguinness and F. van Harmelen, "Owl web ontology language overview," February 2004.
[11] T. Berners-Lee, "Primer: Getting into RDF and semantic web using N3," 2005.
[12] V. Calliaghan, G. Clarke, M. Colley, and H. Hagras, "Embedding intelligence: Research issues for ubiquitous computing," 2006.