Merging Robotics and AAL Ontologies: The RAPP Methodology

Emmanouil G. Tsardoulias[1], Cezary Zieliński[2], Włodzimierz Kasprzak[2], Sofia Reppou[3], Andreas L. Symeonidis[1], Pericles A. Mitkas[4], and George Karagiannis[3]

[1] Centre of Research & Technology – Hellas
6th km Xarilaou – Thermi, 57001, Thessaloniki, Greece
etsardou@iti.gr, asymeon@iti.gr
[2] Institute of Control and Computation Engineering, Warsaw University of Technology
ul. Nowowiejska 15/19, PL-00-665 Warszawa, Poland
C.Zielinski@elka.pw.edu.pl, W.Kasprzak@elka.pw.edu.pl
[3] Ormylia Foundation, 63071 Ormylia – Chalkidiki
sreppou@gmail.com, g.karagiannis@artdiagnosis.gr
[4] Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
mitkas@auth.gr

Abstract. Cloud robotics is becoming a trend in the modern robotics field, as it became evident that true artificial intelligence can be achieved only by sharing collective knowledge. In the ICT area, the most common way to formulate knowledge is via the ontology concept, where different meanings connect semantically. Additionally, a considerable effort to merge robotics with assisted living concepts exists, as the modern societies suffer from lack of caregivers for the persons in need. In the current work, an attempt is made to merge a robotic and an AAL ontology, as well as utilize it in the RAPP Project (EU-FP7).

Keywords: robotics, ontology, AAL, RAPP, cloud robotics, KnowRob, social robots.

1 Introduction

The most promising trend in future robotics is the "collective robotic intelligence", i.e. creating ways for robotic devices to gain knowledge from central (or distributed) repositories, as well as incorporating new concepts deriving from their everyday functionality. Nowadays, numerous projects perform research on this problem, some of which are RoboEarth [1], RoboBrain [http://robobrain.me/#/] and DAvinCi [2], often referred to as Cloud Robotics.

In these projects, robotic systems are capable of storing and distributing knowledge, as well as deriving and inferring information from a variety of sources. These may be stimuli observed by robots, the internet, or even predefined knowledge databases explicitly created for robotic employment.

R. Szewczyk et al. (eds.), *Progress in Automation, Robotics and Measuring Techniques*,
Advances in Intelligent Systems and Computing 351, DOI: 10.1007/978-3-319-15847-1_28

So why all this effort to provide robots the means of learning the world, instead of letting them collect that knowledge by themselves is being made? The truth is that in terms of computation and information inference, a human being is far more advanced in comparison to a robot. This means that tasks trivial to humans need a lot of detail and effort for a robot to perform them. An example is the process of fetching a glass of orange juice, which is extremely complex for a robot, as it does not have the knowledge and experience of a human being, thus several facts are not assumed. The robot has to employ a large skill and algorithm set in order to succeed at this task, such as semantic "translation" of the command, localization or mapping for finding the kitchen, image recognition to detect the glass and serious manipulation skills in order to appropriately handle the glass while the juice is poured. It is apparent that robots need a large amount of structured information in order to perform even trivial tasks and this is the reason behind the latest boost in the existence of "collective robotic intelligence" or "cloud robotics" projects.

RAPP project [3] (Robotic Applications for Delivering Smart User Empowering Applications) is devoted to building a cloud applications and services repository that can be utilized by diverse robots, assisting people with a range of disabilities. Obviously, a single robot cannot have the necessary storage resources to contain all data and procedures required to execute the tasks its user might request. If it lacks some procedures or data, the robot can request and obtain them from the cloud repository either through downloading or by directly executing them on the cloud. Optimally, the user's requests should be formulated in a natural human-like way, thus assuming that robot has the common sense knowledge humans usually possess. This implies that the repository must contain a database where the knowledge is adequately structured as an ontology. Current work focuses on ontologies concerning Ambient Assistive Living, as the robotic assistants will coexist with humans, thus they will have to share the same environmental common sense knowledge. The project's software results will be tested on companion robots preventing people from social exclusion. One such task will focus on improving the confidence level of the person. Many people avoid going out simply because, due to mild cognitive impairment, are afraid that they would forget to shut the windows, turn off the lights, lock the doors, close the valves on the water-taps etc. Checking those hazards can be performed by a companion robot that can report any hazard directly to the person or the caregivers.

2 Application of Companion Robots to Social inclusion

Social inclusion according to Cappo [4] requires a society where all people feel valued, their differences are respected, and their basic needs are met so they can live in dignity. Social exclusion, on the other hand, leads people to isolation as they are banned from the social, economic, political and cultural systems, defining a person's integration into the community [5].

Elderly, facing the threat of losing their independence due to aging and its consequences (health, cognitive, social and financial), are in high risk of social exclusion in the form of deprivation from activities that offer joy, fulfillment and sense of belonging. These can lead to frustration, depression and health decline. Developing policies to confront and prevent social exclusion has been a major concern for the World Health Organization and all European countries. There have been numerous projects suggesting actions and methods to enhance social inclusion of the elderly and to advance their well-being. The information and communication technologies utilization is involved in many of these projects, promoting solutions to support integrated healthcare services in elderly homecare, memory and cognitive enhancement through computer-based exercises and support of social activities. In recent years, a new approach has been introduced to confront social exclusion with the help of assistive robots that can be employed by elderly without intensive training or physical effort and could enable them to gain confidence and achieve autonomous living. Moreover, their humanoid appearance gives a sense of having a companion rather than a machine and decreases loneliness and social deprivation [6–8].

RAPP project suggests a number of applications enhancing independent living of elderly and is using NAO[1], a 58-cm tall humanoid robot by Aldebaran (Softbank Group) that moves, recognizes users, hears and speaks in order to test these applications in real environments. One of the proposed applications and the first to be tested is "Hazard Detection" and consists of a checking overview around the house, aiming to detect possible hazards like an open front door or window, an open fridge door or forgotten switched on lights and oven. The robot can execute this checking procedure on demand or in predefined timestamps, securing that seniors are safe from danger. An open door or window can give access to burglars and put in risk the life of the elderly; an open door of the fridge can lead to food deterioration and consequently to food poisoning; switched on lights or oven can lead to short circuits and fire risks with fatal results. Forgetfulness is a common characteristic of all people when aging. In some cases, forgetfulness is getting more severe and a number of elderly is diagnosed with Mild Cognitive Impairment (MCI), an intermediate stage between the expected cognitive decline of normal aging and the more serious decline of dementia. Forgetting things like open doors or windows and switched on lights, can lead the elderly to lose confidence thus avoid leaving the house or becoming anxious about possible hazards even when inside. Conclusively, they become isolated because of their anxiety and lack of confidence and finally excluded from social context, which in turn deteriorates their health and cognitive abilities in a vicious cycle. Family can also notice their forgetfulness and worry about the safety of the elderly suggesting residential or home care. The RAPP "Hazard Detection" application, performed by the robot, can provide a sense of safety and security both to seniors and their family, thus prolonging an active and independent living for the elderly. Seniors will feel confident to leave the house as the robot deploys hazard detection and informs them, or relax in the house feeling safe as the robot tracks any hazard risks and alert them.

Apart from hazard detection, applications will be developed to assist elderly connecting with family and friends through Skype calls and emails, create their own memory ball with all memories they like to treasure, enhance their attention through cognitive games, and many more applications, using the RAPP platform.

---

[1] `http://www.aldebaran.com/en/humanoid-robot/nao-robot`

3 State of the Art

The current chapter's aim is to provide the basic information about the available ontologies and knowledge repositories existent and utilized by robots or computers in general. Initially, some general purpose ontologies will be described followed by the robotic and AAL oriented ontologies and finally a few online general information sources / repositories will be presented.

3.1 Robotic Ontologies

As described in the introduction, most of the knowledge existent on the web is not directly applicable by a robotic device, as several spatiotemporal and conceptual details are missing. This is the reason behind several efforts on creating a machine readable knowledge container (i.e. an ontology) that a robot can actually utilize, without the need of manually breaking down certain tasks by the user.

One of the most important projects aiming towards this goal is RoboEarth[2] and the robotic knowledge it contains, KnowRob [9]. KnowRob is described as "Knowledge processing for robots" and combines heterogeneous knowledge sources for enriching its database, as well as numerous tools to process and manipulate this knowledge, such as deterministic and probabilistic reasoning mechanisms, clustering, classification and segmentation algorithms, as well as query interfaces and visualization tools. KnowRob extracts knowledge from four distinct sources: a) basic encyclopedic knowledge about the types of things and common-sense knowledge about the household, b) knowledge extracted from various internet sources, c) information derived from robot observation of humans and d) knowledge extracted from the robot's interaction with the environment.

Another robotics-oriented ontology is ORO – The OpenRobots Common Sense Ontology [10]. This ontology is mainly based on the OpenCyc ontology and shares most of its concepts with the KnowRob ontology. This project focuses on the creation of a framework for robots to share a common representation of environmental concepts.

Finally, the Proteus project ("Plateforme pour la Robotique Organisant les Transferts Entre Utilisateurs et Scientifiques" or Robotic Platform to facilitate transfer between Industries and academics)[3] has as goal to bridge the French robotic community with the industry, in order to enable knowledge transfer between the two different fields.

3.2 AAL / ICT Ontologies

The need for knowledge organization in a computer comprehensible way is apparent for almost any scientific field. The same applies for the social/assisted living paradigms. Their aim is to formally specify the context of things and actions observed in

[2] http://roboearth.org/
[3] http://www.anr-proteus.fr/

AAL, in order to make robots and machines context-aware, thus able to sense and react based on their environment. In [11], context may be characterized as:

- Static or dynamic: Static context describes unaltered facts, such as a person's birthdate. A dynamic paradigm may be a person's age, as in increases with time.
- Imperfect: The information stored may be altered in short notice or the context sources may provide erroneous data.
- Alternative representation: This describes the ambiguity between multiple representations for a single concept.
- Highly interrelated: The contexts of entities have strict (or relaxed) relations between them, fact that includes information as well.

Next, some examples of projects developing ALL/ICT ontologies are briefly described.

The first case is SOPRANO (Service-Oriented Programmable Smart Environments for Older Europeans) [12]. SOPRANO is not just an ontology, but a combination of ontology-based techniques and a service-oriented architecture. SOPRANO comprises three main parts: the SOPRANO ontology, the service-oriented infrastructure and the ambient middleware.

Another Ambient Assisted Living project is SINDI [13], whose name derives from the words Secure and Independent Living. This is an intelligent home healthcare system containing semantic, inference and reasoning mechanisms, in conjunction to a wireless sensor network. Its main functionality concerns the constant acquisition of new semantic data.

VAALID[4] (Accessibility and Usability Validation Framework for AAL Interaction Design process) is an FP7 – ICT project, providing an ontologies set, aiming at modeling an AAL environment and its contents (actors, spaces, devices). ELDeR [14] is an ontology, aiming in facilitating the procedure of creating ambient intelligent systems (AmI) and containing context information relative to the elder's daily activities. Additionally, its inspiration is to enable developers to create systems which can infer potential future risks. Finally, OpenAAL [15] (the open-source semantic middleware for ambient assisted living) is an open source software project based on the SAM (SOPRANO Ambient Middleware) architecture and implementation, oriented towards AAL scenarios.

Next, the RAPP ontology design selections are presented and some analytical examples of their basic entities are described.

4 Ontology Design

Since RAPP is a multidisciplinary project containing elements both from robotics and assisted living fields, it is crucial to employ ontologies that comprise semantical information from both areas. Some of the prerequisites before the actual ontologies selection were:

---

[4] http://www.vaalid-project.org/project-details.html

– The tools should have ROS[5] support, as the entire RAPP platform will be ROS-based. This will boost the modules' interconnection capabilities, as every node will have seamless access to the knowledge database.
– Both ontologies should have a common file format for the information to be easily joined.
– The ontologies should be state of-the-art, in order to fully take advantage of their capabilities and to make sure that adequate support from the community exists.

4.1 KnowRob

Taking these prerequisites under consideration, we selected KnowRob as the robotic ontology to be utilized. As stated before, KnowRob extends classical ontologies by containing information vital for a robot's operation, such as spatial/mathematical data (coordinates, matrixes, vectors etc), as well as temporal (events, timeslots and others).

The main reason behind KnowRob's selection was that it is considered state-of-the-art in robotic ontologies. Additionally, it is a software product of the RoboEarth project which is in many parts similar to RAPP, like cloud robotics, knowledge storage and distribution and knowledge extraction from different sources. This choice was made having in mind that utilizing already existent and tested tools would boost performance and promote collaboration between diverse scientific groups. Furthermore, an obvious advantage in utilizing KnowRob is its compliance to ROS, as any developer will be able to query the ontology through a ROS node. This is extremely helpful in keeping the proposed overall system architecture simple.

It should be mentioned that in KnowRob all knowledge is implemented in OWL (Web Ontology Language) representation. OWL files are XML-like files, allowing to formally describing relational knowledge, i.e. semantically interconnected concepts. Also, as stated earlier, KnowRob is implemented in SWI-Prolog. Prolog is used to load, store and reason on the knowledge contained in the OWL files. The ROS wrapper was created by utilizing the JPL interface (Java/Prolog bidirectional Interface) and the experimental (up until now) rosjava package. For a more detailed KnowRob documentation, many examples / tutorials can be found in the official KnowRob website. The KnowRob upper taxonomy includes the following concepts (among others):

– `Thing`: The most basic entity of the ontology. It is the most abstract concept and all other classes derive from it, i.e. everything is a "thing". Its four subclasses are:
  - `TemporalThing`: Describes all elements related time. Its subclasses are
    - `TimeInterval`: (for example time points, dates, time of the day)
    - `Situation`: For example a posture or a grasp
    - `Event`: A general concept of an event. This may contain:
      - `SensoryEvent` (e.g. perception of an object by camera or distance sensors)
      - `StateChangeEvent` (e.g. heating, freezing, vaporization)
      - `Action`: Elements that have to do with time and performing an action.

[5] `http://www.ros.org/`

- `Agent-Generic`: Used to describe robots as agents.
- `SpatialThing`: Describes things physically existent in an environment:
  - `Map` of an environment
  - `Place`: A relevant place in the environment (e.g. kitchen)
  - `SurfaceRegion`: Different sides of objects (right side, front, back)
  - `EnduringThing-Localized`: All objects that can be assigned a location, These can be:
    - `EmbodiedAgent`
    - `PhysicalDevice`: e.g. tools
    - `Connection-Physical`: Any joint
    - `HumanScaleObject`: Objects related to humans.
- `MathematicalOrComputationalThing`: Mathematical concepts
  - `Units`
  - `MathematicalObjects` e.g. Vectors, Matrices etc.

It must be noted that KnowRob provides a basis for the ontology a robot needs to operate in a household and specifically concerning the "assistive kitchen project" concept, as stated in the Moritz Tenorth's dissertation [10], the main KnowRob creator and contributor. Thus, it is understandable that we will develop any extensions needed to be seamlessly employed in a household environment, towards the user scenarios described.

4.2 OpenAAL

In order to assess the social ontology problem, OpenAAL will be employed. As stated, OpenAAL is the result of the SOPRANO integrated project, representing a flexible and powerful middleware for AAL scenarios. OpenAAL provides a full integrated platform for ambient assisted living, containing not only semantic storage and inference, but many tools for user context management, multi-paradigm context augmentation, context aware behaviors and others. Though, for this project's needs only the ontology will be utilized. The high level ontology is contained in an owl file, thus it can easily be merged to the KnowRob ontology. An example of the high level ontology tree is presented below:

- `OpenAALThing`: The top level concept. Every other concept / class is an `OpenAALThing`.
  - `Activity`: Describes a person's activity
  - `Acute-State`: Describes a "dangerous" state for AAL, such as a fall.
  - `Appointment`: A person's appointments.
  - `Device-State`: Describes the state of a device (electronic). This class contains:
    - `Open-State`: A device's physical state; e.g. if an oven is open or closed.
    - `Power-State`: A device's electrical state; e.g. if an oven is on, off or standby.

- `Due-State`: Due state can either be Due or Not-Due and can be related to an appointment to define whether the appointment is due or not due.
- `Locatable-Entity`: Every object that can be located in a household. This class contains:
  - `Device`: Examples are doors, lamps, TV, couch, oven etc.
  - `Low-Level-Thing`: Describes actuators and sensors.
  - `Person`: Class that differentiates between AP (Assisted Person) and Carers.
- `Location`: Contains Rooms and Room-Parts. Rooms comprise the Bathroom, Kitchen and Unspecified-Room classes.
- `Message`: An abstract class that contains:
  - `Answer`: Its subclasses are Confirmation, Yes and No.
  - `Information`: Contains Alarms (like Fall-Detected-Message), Notifications and Reminders.
  - `Question`: Its only subclass is Yes-No-Question
- `Message-Modality`: Contains messages that need to be assessed. Its two subclasses are Sound-Message and Text-Message.

It is apparent that the OpenAAL ontology provides the basis for semantically describing AAL scenarios. Nevertheless it is not complete, i.e. it does not contain every class to be found in a household environment, such as the different electric appliances, locations or devices. Thus, this ontology will be enriched with the necessary classes and instances needed to test the user scenarios prescribed in the RAPP project. The second fact to be mentioned is that OpenAAL and KnowRob contain some classes similar to a certain extend. Some examples are the `KnowRob:Thing` and `OpenAAL:OpenAALThing`, the `KnowRob:PhysicalDevice` and `OpenAAL:Device` etc. Taking this fact under consideration we decided to conceptually merge the two ontologies where possible and afterwards perform the necessary extensions. Next, the actual ontology implementation and employment within the RAPP system will be described.

5 RAPP Ontology Employment

5.1 Setup

As evident from the KnowRob description, its primary goal is to semantically integrate the ensemble of possible operations that can occur in a household, regarding the capabilities and perception of robotic devices operating within it. This fact suggests that the "natural environment" for the KnowRob's deployment is a single household, where the spatiotemporal information of the robots and objects are stored and tracked. On a different basis, this suggests that a centralized system in which KnowRob will be installed, will exist within the household where the robots operate. Unfortunately, this is not the case in RAPP.

As described, our main concern is not to perform severe modifications in a household, and the absolutely needed modifications not to burden financially the end users. So, simple modifications like room and device tagging using QR tags can be applied, but installing a personal computer in order to enable heavy duty local operations is out of the question. Having said that, the only place where KnowRob could be installed is in the robotic devices' computational units. Nevertheless, the main problem regarding this decision is that we intend to create a, as more as possible, a globally applicable solution, i.e. for any robotic device that runs a Linux operating system. Of course not all robots have large storage and computational abilities, an example of which is NAO.

Our way to overcome these problems is to install KnowRob and OpenAAL on the RAPP cloud part. This decision holds some advantages and disadvantages; the main disadvantage is that KnowRob isn't utilized the way it was created for, meaning that a large percent of its capabilities will remain dormant. Of course this infers that we must "invent" a new employment way for ontology. On the other hand, this solution enables us to create a centralized knowledge repository, from which all the robots can benefit, as described in the RAPP Description of Work.

5.2 Access

Both OWL files (KnowRob-common and OpenAAL ontologies) will be merged and loaded in the KnowRob software, in order to be able to perform semantic queries. As aforementioned, KnowRob is implemented in SWI-Prolog, thus it is hard to access semantic information from a C++ ROS node. Fortunately, the developers of KnowRob provide a KnowRob ROS stack enabling the queries to the ontology via a ROS service. The operations to be performed in the RAPP Platform (containing the RAPP Store, the RIC – RAPP Improvement Centre and the HOP services – Fig. 1) will be developed in the form of ROS nodes, as this is a tested way to create a closed system with intra-process communication. The robot–cloud communication is implemented in HOP [16, 17]. Each ROS node providing services to the outside world will be wrapped with a thin HOP layer, which in turn uptakes the task of communicating with the Robotic Applications (RApps) that are developed in HOP and exist in the robots, after downloading them from the RAPP Store. Thus, each robot can have access to the knowledge repository by utilizing the provided HOP services. HOP is essentially the glue that holds the whole system together. HOP is a multitier programming environment, built on top of web protocols and languages, orchestrating data and commands transfer among objects such as web services and user interface components. HOP is typically used to coordinate home automation and robotic environments for assisted living.

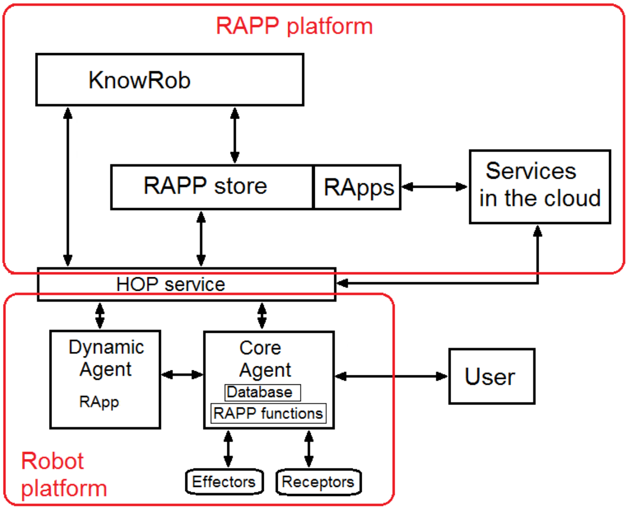

Fig. 1. The main actors in RAPP

5.3 Utilization

This section describes the means of the ontology utilization within the RAPP boundaries. In a few words, after the ontology is expanded with the necessary classes, it is employed as inference tool for the semantic information, by creating anonymized class instances from the objects and actions observed by each robot and performing machine learning algorithms on them. On the other hand, the eponymous information is stored in a secure database, where a ROS node has access. The concept behind these decisions is made apparent by presenting a RApp example. The following actors are distinguished:

1. User A– the human user;
2. CoreAgent – located on the "Robot platform" (i.e. "the robot") that controls the robot's effectors and receptors, manages a local short-term memory (i.e. the robot's Database, which stores short-term data, such as navigation maps, needed object models, own state and also contains some Rapp Functions for image processing and navigation);
3. DynamicAgent: current RApp running on the "Robot platform" ;
4. HOPservice: communicates the robot (both agents) with the RAPP platform;
5. RAPPplatform: the cloud of RAPP resources (e.g. services like SpeechRecognition or ObjectRecognition);
6. RAPPstore: it holds the RApps ;
7. KnowRob: manages an Ontology database with a SWI-Prolog-program – allows communication with the Ontology – located on the "RAPP platform";
8. SpeechRecognitionService: a service application in the RAPP cloud;
9. ObjectRecognitionService: a service for object recognition in images;

Step I. The robot gets a command from the user A, who asks it to “search for hazards”, and it recognizes it with the help of the RAPPplatform. Implementation:

- UserA issues a voiced command ⇒ the CoreAgent creates an audio file with the command, e.g. “search for hazards”.
- The CoreAgent calls the RAPPplatform via a HOPservice ⇒ the RAPPplatform notifies it is an audio file and calls a Speech Recognition service.
- The SpeechRecognitionService returns keywords, e.g. “search, hazards”.
- The RAPPplatform checks there exist a “hazardRApp” in the RAPPstore and notifies the CoreAgent via HOPservice.

Step II. The robot downloads, installs and starts the selected RApp as a DynamicAgent.
Step III. The current RApp (in the DynamicAgent) evokes a HOP service searching for the term “hazard” in the ontology. The service returns some associations, e.g. open oven, in order for the robot to search for it.
Step IV. Initially, the robot tries to find out if it has knowledge of the specific user’s oven, i.e. the DynamicAgent queries the Database of the CoreAgent for a model of “oven”. The first time, the robot does not know what an oven is. Hence, no information is stored and the DynamicAgent queries the KnowRob for a model of oven. If no any robot has yet tried to find the oven-related information, there is none in the RAPP. But if there exists a model of “oven” then it is returned to the DynamicAgent and the following steps V-VI are not needed.
Step V. The robot asks the user to show the oven, then it downloads a human tracking RApp and runs on-line a cycle of steps: image capture, audio capture, image analysis, speech recognition, navigation control, map update. This needs a lot of steps dealing with image processing – human pose detection, human arm and head tracking, hand gesture interpretation – with navigation control for human following, and with speech recognition (a human saying something like: “this is an oven.open”). At the end, the robot approaches the oven to acquire a final image of it.
Step VI. The human tracking RApp is finished and the DynamicAgent returns to run the ancestor RApp. The image of an open oven is captured and the DynamicAgent calls an ObjectRecognitionService for image feature extraction model creation. Then two operations are called by the DynamicAgent to complete the model registration: an instance is created in the ontology, under the “oven.open” class, a subclass of “oven”, containing the oven’s visual features and a registration is created in the database that connects the specific features to the specific user.
Step VII. Now the DynamicAgent has identified the oven model and it navigates and executes machine vision algorithms in order to search for an oven and to classify its state (open or closed). Eventually it triggers an alarm, playing back an audio message, e.g. “Attention: open oven”.
Step VIII. The DynamicAgent continues searching for other hazards (repeating from step IV, e.g. hazards = “lights.on”) or it finishes the hazard RApp.

5.4 Conclusions

In this document, the RAPP approach towards the utilization of ontologies is described. The ontology employment will play a crucial role in the Cloud Robotics aspect of the RAPP project, enabling the knowledge collection, processing and distribution among heterogeneous robots with certain common sensors.

Initially, the state of the art was described in the field of knowledge sources available online. As RAPP is oriented towards two different areas – robotics and inclusion or AAL – the main ontology schemes of these two fields were investigated. At first, some general purpose ontologies were presented, attempting to capture semantic information on an encyclopedic scale. Then, the state of the art robotic ontologies were described (the most important of which is KnowRob), as well as the AAL/ICT ones. Finally some general information sources were presented, which are known to have been used in conjunction with robots by some scientific groups.

The third chapter contains the ontology design, i.e. which ontologies were selected for employment from each field and the reasons beyond this choice. As stated there, the selected ontologies were KnowRob, concerning robotics and OpenAAL concerning AAL applications.

Finally, the last chapter provides an insight into the actual ontology setup, means of access and utilization within the RAPP project. Additionally, an intuitive example was given, in order to conceptually present and support the aforementioned choices.

The main drawback of the presented implementation is that the KnowRob platform is not utilized in its full power, as it cannot be installed locally within each household. This can be avoided in future versions of RAPP, where additional tools will be supported, such as mobile phones, tablets or even PCs. If so, a future extension of the system may include local KnowRob instances that provide both specific spatiotemporal information to the robots that operate in the house, and generalized high level knowledge to a cloud instance of the RAPP Platform.

Acknowledgements. Parts of this work have been supported by the FP7 Collaborative Project RAPP (Grant Agreement No 610947), funded by the European Commission.

References

1. Waibel, M., Beetz, M., Civera, J., D'Andrea, R., Elfring, J., Galvez-Lopez, D., Haussermann, K., et al.: A World Wide Web for Robots. IEEE Robotics & Automation Magazine (2011)
2. Arumugam, R., Reddy Enti, V., Bingbing, L., Xiaojun, W., Baskaran, K., Foo Kong, F., Senthil Kumar, A., Dee Meng, K., Wai Kit, G.: DAvinCi: A cloud computing framework for service robots. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 3084–3089. IEEE (2010)

3. Psomopoulos, F., Tsardoulias, E., Giokas, A., Zielinski, C., Prunet, V., Trochidis, I., Daney, D., Serrano, M., Courtes, L., Arampatzis, S., Mitkas, P.A.: RAPP System Architecture. In: Proc. of the Assistance and Service Robotics in a Human Environment, Workshop in conjunction with IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago (September 2014)
4. Cappo, D.: Social inclusion, participation and empowerment. ACOSS Info Papers, p. 6 (2003)
5. Barrios-Aranibar, D., Gurgel, V., Gonçalvez, L.M.G., Santos, M., Araújo, G.R., Roza, V., Nascimento, R.A.: Technological inclusion using robots. In: Anais do II ENRI-Encontro Nacional de Robótica Inteligente, vol. 7. XXVI Congresso da Sociedade Brasileira de Computaçao-SBC2006, Campo Grande (2006)
6. Roupa, Z., Nikas, M., Gerasimou, E., Zafeiri, V., Giasyrani, L., Kazitori, E., Sotiropoulou, P.: The use of technology by the elderly. Health Science Journal 4(2) (2010)
7. Dautenhahn, K., Woods, S., Kaouri, C., Walters, M.L., Koay, K.L., Werry, I.: What is a robot companion-friend, assistant or butler? In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 1192–1197. IEEE (2005)
8. Scherer, M.J., Glueckauf, R.: Assessing the Benefits of Assistive Technologies for Activities and Participation. Rehabilitation Psychology 50(2), 132 (2005)
9. Tenorth, M., Beetz, M.: KnowRob—knowledge processing for autonomous personal robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, pp. 4261–4266. IEEE (2009)
10. Lemaignan, S., Ros, R., Mosenlechner, L., Alami, R., Beetz, M.: ORO, a knowledge management platform for cognitive architectures in robotics. In: 2010 IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS), pp. 3548–3553. IEEE (2010)
11. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: Mattern, F., Naghshineh, M. (eds.) PERVASIVE 2002. LNCS, vol. 2414, pp. 167–180. Springer, Heidelberg (2002)
12. Wolf, P., Schmidt, A., Klein, M.: SOPRANO-An extensible, open AAL platform for elderly people based on semantical contracts. In: 3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI 2008), 18th European Conference on Artificial Intelligence (ECAI 2008), Patras, Greece (2008)
13. Mileo, A., Merico, D., Pinardi, S.: Bisiani. R., A logical approach to home healthcare with intelligent sensor-network support. The Computer Journal (2009)
14. Saldaña-Jimenez, D., Rodríguez, M.D., García-Vázquez, J.-P., Espinoza, A.-N.: ELDeR: An ontology for enabling living inDependently of risks. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 622–627. Springer, Heidelberg (2009)
15. Wolf, P., Schmidt, A., Otte, J.P., Klein, M., Rollwage, S., König-Ries, B., Dettborn, T., Gabdulkhakova, A.: openAAL-the open source middleware for ambient-assisted living (AAL). In: AALIANCE Conference, Malaga, Spain, pp. 1–5 (2010)
16. Serrano, M., Gallesio, E., Loitsch, F.: Hop: a language for programming the web 2.0. In: OOPSLA 2006: Companion to the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications, pp. 975–985. ACM (2006)
17. Serrano, M., Berry, G.: Multitier programming in hop – a first step toward programming 21st-century applications. Communications of the ACM 55(8), 53–59 (2012)