

A Knowledge Management System for Assistive Robotics

Luigi Asprino^{1,2}, Aldo Gangemi^{1,3} Andrea Giovanni Nuzzolese¹,
Valentina Presutti¹, and Alessandro Russo¹

¹ STLab, ISTC-CNR, Rome, Italy
`{name.lastname}@istc.cnr.it`

² DISI - Università di Bologna, Bologna, Italy

³ LIPN, Université Paris 13, Sorbone Cité, UMR CNRS, Paris, France

Abstract. In this paper we demonstrate how to use an ontology network in order to fill the gap between knowledge and robots' abilities. The demonstration is focused on two components of the knowledge management system of MARIO robots. These components are the MARIO Ontology Network (i.e. MON) that organises knowledge in MARIO and an Object-RDF mapper, called Lizard, that dynamically generates APIs on top of the MON to enable the interaction between software components that implement robot's abilities and the MON itself.

1 Introduction

The MARIO robot⁴ is an assistive robot that has to support a set of knowledge-intensive tasks aimed at (i) helping patients affected by dementia to feel more autonomous and less lonely, (ii) supporting carers in their activity to assess the patient's cognitive condition. Examples of knowledge-intensive tasks are the Comprehensive Geriatric Assessment (CGA) and the triggering of appropriate entertainment activities. In order to enable this tasks MARIO features a set of abilities implemented by pluggable software components. MARIO abilities, when executed, contribute to and benefit from a common knowledge base, which is modelled according to an ontology network called MARIO Ontology Network (MON). In this paper we present two components of the Knowledge Management System of MARIO robots, i.e. (i) the MARIO Ontology Network (MON) and (ii) Lizard. The main objective of MON is to provide MARIO robots with the needed infrastructure to organise knowledge. Instead, the main objective of Lizard is to provide MARIO robots with APIs dynamically built on top of MON in order to allow the software components implementing the abilities to access, query and store the knowledge organised by the MON. The rest of the paper is organised as follows: (i) Section 2 presents the related work; (ii) Section 3 provides details about the design methodology adopted for modelling the MON (cf. Section 3.1) and the knowledge areas covered by the ontology network (cf. Section 3.2); (iii) Section 4 describes Lizard; finally, (iv) Section 5 provides the conclusions.

⁴ <http://www.mario-project.eu/portal/>

2 Related work

In recent year many ontology based knowledge systems for robotics have been proposed. Examples are ORA [6], RACE [8], and OUR-K frame work [4]. ORA introduces a core ontology that encompasses a set of terms commonly used in Robotics and Automation along with the methodology we have adopted. RACE uses an ontology for connecting actuator and sensory experiences with semantics for implementing robots' abilities. OUR-K presents an ontology-based unified robot knowledge framework that integrates low-level data with high-level knowledge for robot intelligence. To the best of our knowledge none of the cited works uses a modular ontology network to organise knowledge and foster its reuse for implementing robots' abilities by means of an Object-RDF mapper that dynamically provides HTTP REST APIs for interacting with such an ontology network.

3 MARIO Ontology Network

In the following sections we describe the methodology adopted for designing the MON and the resulting ontology modules.

3.1 Design Methodology

The MON is designed by following best design practices and pattern-based ontology engineering aimed at extensively re-using Ontology Design Patterns (ODPs) [3] for modelling ontologies. The design methodology that we followed is based on an extension [7] of the eXtreme Design [2], an agile design methodology developed in the context of the NeON project⁵. Such an extension mainly focuses on providing ontology engineer with clear strategies for ontology re-use. According to the guidelines provided by [7], the strategy we adopted for the development of the MON is the *indirect re-use* of ontology design patterns and alignments. This means that ODPs are used as templates. At the same time, the ontology guarantees interoperability by keeping the appropriate alignments with the external ODPs, and provides extensions that satisfy more specific requirements. With this type of reuse, the potential impact of possible changes in the external ontology modules (e.g. ODPs) is minimised.

3.2 Ontology Network Modules

Based on this methodology we designed the MON as a networked ontology⁶ composed of different modules that cover different knowledge areas that are relevant to MARIO in order to make it a cognitive agent able to support older

⁵ <http://www.neon-project.org/nw/>.

⁶ The ontologies part of the network are available at <http://www.ontologydesignpatterns.org/ont/mario/>. The root of the network is available at <http://www.ontologydesignpatterns.org/ont/mario/mario.owl>.

patient affected by dementia. The knowledge areas were identified by analysing the use cases emerged from the system specification carried on for Pilot 1 in the context of the Work Package 1 [1] of the MARIO project. These use cases mainly describe actions and behaviours that the robot should perform or select. However, they also provide us with detailed descriptions about the nature of the knowledge that the robot should deal with in order to perform and select actions and behaviours, respectively. For each use case we highlighted the knowledge domains required to address the use cases. This process was enabled by the identification of the *competency questions* from the textual descriptions of the use cases. We remark that in knowledge engineering the competency questions are commonly identified as the requirements that an ontology has to address. Hence, the knowledge domains emerged from the competency questions we collected, i.e. the knowledge topics that specific competency questions answers to. Finally, we gathered a set of top-level knowledge areas by iteratively generalising the knowledge domains.

Figure 1 shows the ontologies composing the MON. Each color identifies a specific knowledge area, namely:

- **Personal sphere.** People information, information about relationship among people, contacts, etc.;
- **Life events.** Information about everyday events, patterns memories, scheduling, plans, etc.;
- **Social and multimedia content.** Online social network community, multimedia content such as photos, videos, movies, documents;
- **Environment.** Information about rooms, furnitures, objects, etc.;
- **Health sphere.** Information about living patterns, health patterns, vital signs, anything related to CGI;
- **Emotional sphere.** Information about emotions, sentiments, interests, opinions related to people etc.;
- **Open knowledge.** Speech-derived data, web-extracted data, etc.;
- **Regulatory sphere.** Information about norms, rules, social habits etc.;
- **“MARIOception”.** Information about the MARIO robot, its functionalities, and the applications it is able to run and to do.

4 Binding MON with robots’ abilities

The MON organises knowledge in MARIO robots according to the areas described in Section 3.2. However, robots’ abilities are implemented in MARIO by software components that can be plugged-in by means of the REST architectural style. These software components are supposed to interact with the MON in order to access, store, and dealing with the knowledge organised by the MON. Hence, MARIO is provided with a Knowledge Management System that relies on an Object-RDF mapper called *Lizard*⁷. An Object-RDF mapper is a system

⁷ Lizard is available as open source with Apache Version 2 license on GitHub at <https://github.com/anuzzolese/lizard>

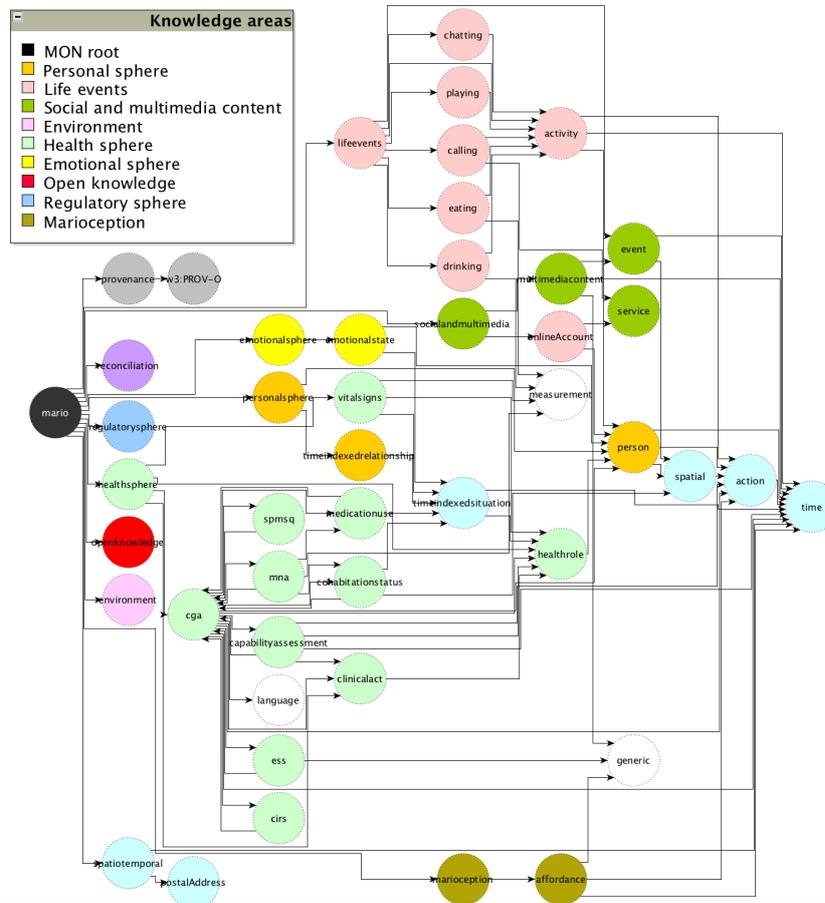


Fig. 1: Core modules of the MON. Circles and arrows represent ontologies and `owl:import` axioms, respectively.

that exposes the RDF triple sets as sets of resources and seamlessly integrates them into the Object Oriented paradigm. However, differently from existing systems such as SuRF⁸ or ActiveRDF⁹ [5], Lizard provides a RESTful layer that exposes Object Oriented paradigm by using the REST architectural style over HTTP. Basically, Lizard dynamically generates Java and HTTP REST APIs from the MARIO Ontology Network (MON). Those APIs reflect the semantics of MON and allow transparent access to the knowledge base. This means that a client application (i.e., any component of the MARIO robotic framework) can access the knowledge base without any prior knowledge of the ontologies used within the KMS. For example, this avoids client applications to deal with OWL

⁸ <https://pythonhosted.org/SuRF/>

⁹ <https://github.com/ActiveRDF/ActiveRDF>

and RDF or to interact with a knowledge base by means of SPARQL queries. Additionally, Lizard embeds an Access Control Management System (ACMS) that enables the setup of specific access policies in order to guarantee the access (either in read or write mode) to specific knowledge areas only to a set of allowed entities/systems. For example, an application that performs some entertainment activity (e.g. play music) would not be allowed to access knowledge about the Continuous Geriatric Assessment (CGA). Hence, the ACMS allows Lizard to deal with some important data management aspects regarding data access, security and data privacy.

5 Conclusions

In this paper we introduced the MARIO Ontology Network (i.e. MON) and Lizard that are two components of the knowledge management system adopted by MARIO robots. These two component enable (i) modular knowledge organisation and (ii) dynamic APIs generation in order to enable the development of intelligent abilities in MARIO robots.

Acknowledgements. The research leading to these results has received funding from the European Union Horizons 2020 the Framework Programme for Research and Innovation (2014-2020) under grant agreement 643808 Project MARIO "Managing active and healthy aging with use of caring service robots".

References

1. Bisset, D.L.: D1.1 MARIO System Specification for Pilot 1. Tech. rep., MARIO consortium (dec 2015)
2. Blomqvist, E., Presutti, V., Daga, E., Gangemi, A.: Experimenting with eXtreme Design. pp. 120–134. Springer (2010)
3. Gangemi, A., Presutti, V.: Ontology Design Patterns. pp. 221–243. Springer (2009)
4. Lim, G.H., Suh, I.H., Suh, H.: Ontology-based unified robot knowledge for service robots in indoor environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41(3), 492–509 (2011)
5. Oren, E., Heitmann, B., Decker, S.: ActiveRDF: Embedding Semantic Web data into object-oriented languages. *Web Semantics* 6(3), 191–202 (2008)
6. Prestes, E., Carbonera, J.L., Rama Fiorini, S., Jorge, V.A.M., Abel, M., Madhavan, R., Locoro, A., Goncalves, P., Barreto, M.E., Habib, M., Chibani, A., Gérard, S., Amirat, Y., Schlenoff, C.: Towards a core ontology for robotics and automation. *Robotics and Autonomous Systems* 61(11), 1193–1204 (2013)
7. Presutti, V., Lodi, G., Nuzzolese, A.G., Gangemi, A., Peroni, S., Asprino, L.: The role of ontology design patterns in linked data projects. pp. 113–121. Springer (2016)
8. Rockel, S., Neumann, B., Zhang, J., Dubba, K.S.R., Cohn, A.G., Konečný, Š., Mansouri, M., Pecora, F., Saffiotti, A., Günther, M., et al.: An ontology-based multi-level robot architecture for learning from experiences. In: *Designing Intelligent robots: Reintegrating AI II*. pp. 52–57. AAAI Press (2013)