ORO, a knowledge management module for cognitive architectures in robotics

Séverin Lemaignan, Raquel Ros, Lorenz Mösenlechner, Rachid Alami, Michael Beetz

ORO, a knowledge management module for cognitive architectures in robotics

Séverin Lemaignan*†, Raquel Ros*, Lorenz Mösenlechner†, Rachid Alami* and Michael Beetz†
*CNRS - LAAS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31077 Toulouse, France
Email: surname.name@laas.fr
†Intelligent Autonomous Systems, Technische Universität München, Munich, Germany
Email: surname.name@in.tum.de

***Abstract*—This paper presents an embedded cognitive kernel, along with a common-sense ontology, designed for robotics. We believe that a direct and explicit integration of cognition is a compulsory step to enable human-robots interaction in semantic-rich human environments like our houses. The OpenRobots Ontology (ORO) kernel allows to turn previously acquired symbols into concepts linked to each other. It enables in turn reasoning and the implementation of other advanced cognitive functions like events, categorization, memory management and reasonning on parallel cognitive models. We validate this framework on several cognitive scenarii that have been implemented on three different robotic architectures.**

I. Introduction

A robot interacting with humans in everyday life situations needs much knowledge. For instance, if a robot is asked to set a breakfast table, how to choose the right items? Or on the contrary, how to know that an item is odd in this context? To make the required decisions, the robot needs a rich symbolic model of its environment and rules that will allow to reason on its knowledge.

If the robot directly interacts with humans, it may even be necessary to take the human perspective in order to percieve and model the human own beliefs on the world. For instance, suppose there are two different jams on a table, but the human can only see one of them (because a third object occludes the second jam from its view). If the human asks for "the jam", the robot must infer that it is referring to the jam it sees. This ability for the robot to think about other agents' mental states is part of the so-called *theory of mind* [1] and is fundamental for legible and pertinent human-robot interaction.

These challenges require not only to provide robots with perceptual abilities, but also a comprehensive model of the roles, relationships and context of objects in the environment, as well as beliefs and intentions of other agents. Moreover, this understanding must rely on a formal encoding that requires high expressivity while remaining well suited for machine processing in order to be used by the robot.

This paper introduces ORO, an easy-to-deploy, event-oriented, platform for symbolic knowledge storage and reasoning. Based on an ontology, it brings several advanced cognitive features to robotic architectures like categorization or explicit modelling of agents mental states. A common-sense ontology, focused on human-robot interaction needs, is as well presented.

We apply this tool to grounded, symbolic interaction and decision-making in human environements, as a module within larger cognitive robotic architectures.

This open-source knowledge management module and its applications are the main contributions of this paper to the field of cognitive robotics.

We present a brief overview of the current cognitive and knowledge processing approaches within the robotics community in Section II. The current stage of ORO development is introduced in Section III and concrete applications on three different robotic architectures are described in Section V. Section VI concludes the paper.

II. Related work

Pionnering works on questions related to cognition in robotics include papers by McCarthy [2], Sloman et al. [3] or Levesque and Lakemeyer [4]. Most of the challenges of cognitive robotics can be summarized from these three articles.

In the field of symbolic knowledge processing for robots Gunderson and Gunderson [5] introduce the concept of reification (based on both recognition and pre-afference) as an intermediate step between pattern recognition and symbol grounding. Their underlying storage of knowledge relies on ontologies and bio-inspired memory model.

Daoutis et al. [6] also tackle grounded knowledge and common-sense reasoning in their KR&R system. They base their knowledge model directly on the *ResearchCyc* ontology (including the *MicroTheories* concept), used in combination with the CycL language.

Tenorth and Beetz [7] introduce KnowRob, a knowledge processing framework based on Prolog. Its underlying storage is based on an OWL ontology, derived from OpenCyc. They introduce as well the concept of *computable relationship* to compute on request RDF triples describing spatial relations between objects, probabilities for certain actions to occur, etc.

While *computables* enable better scaling (lazy evaluation of relationships), this prevents on the other hand an efficient use of the reasoner to classify and infer new statements since this generally requires at any time the complete set of statements to be available. Inconsistencies in the robot knowledge are as well more difficult to detect. Our work explores a parallel approach where the set of statements is classified *a priori*.

III. ORO, A ONTOLOGY-BASED KNOWLEDGE PROCESSOR

A. Architecture

ORO[1] is an open-source (BSD-like), socket-based server build on the top of a standard RDF triples store. Figure 1 illustrates the overall architecture. A *front-end* accepts and manages connections from client components. The clients' requests are processed by a set of internal *modules*. Besides basic operations like to store and retrieve knowledge, several pluggable *modules* have been developped that add more complex cognitive and human-robot interaction abilities (see below). Ultimately, the modules rely on an ontology *back-end* that is made of one or several independant RDF stores. The knowledge is actually stored in this back-end.

ORO is designed as a service: it can been seen as an intelligent blackboard that allows other modules in the robot to push or pull asserted and infered knowledge.

Knowledge is represented in ORO in first-order logic formalism, as RDF triples (for instance `<robot isIn kitchen>`). ORO relies on a dialect of RDF, OWL[2] Description Logic, which is the decidable part of OWL. The underlying RDF triples storage is the open-source Jena framework[3]. We use it in conjunction with the equally open-source Pellet[4] reasoner to ensure the continuous classification of the storage: as soon as the robot adds a new fact in the knowledge base, this fact is included in the complete set of asserted facts and used for reasoning.

For instance, let us assume that the robot knows that `WaterContainer` is the collection of all the objects that may contain water. And let us consider that it knows about some `cup_1` (`<cup_1 rdf:type Cup>`). If the robot acquires the fact (for instance by asking the human) that a cup is a water container (`<Cup rdfs:subClassOf WaterContainer>`) then it will automatically infer that the `cup_1` can contain water, *i.e.* `<cup_1 rdf:type WaterContainer>`. The inferred statement is dynamically added into the knowledge base.

B. The OpenRobots Ontology

One of the major issues that soon arises when dealing with knowledge representation in human-robot interactions is the lack of *common-sense knowledge*. While difficult to estimate, the common sense knowledge (both declarative –*rain wets*– and procedural –*how to open a door*–) represent a huge part of our everyday knowledge, and the lack of such knowledge by robots is especially frustrating in human-robot interactions.

Several important projects are trying to fill this gap by providing machine-processable (the OPENMIND project[8], for instance) repository of common sense facts produced by humans. These knowledge bases are extremely valuable but

[1]Project homepage: http://www.laas.fr/~slemaign/oro-server

[2]http://www.w3.org/TR/owl2-overview/

[3]http://jena.sourceforge.net

[4]http://clarkparsia.com/pellet/

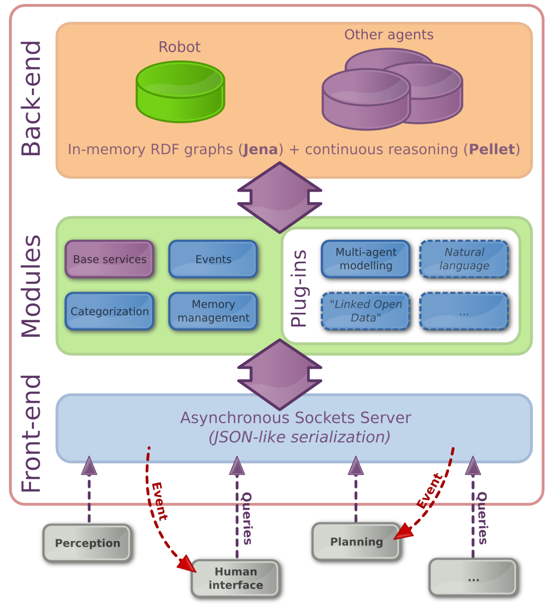


Fig. 1. Overview of the ORO architecture

remain difficult to use in a pervasive way because of both their uncompleteness and the lack of good connections with underlying, unambigous concepts.

The knowledge that the robot acquires (by perception or interaction) needs indeed to be connected to other chunks of knowledge to become actually useful. This requires at least an agreement on common identifiers to symbolize identical concepts. To this end, the ORO server can be loaded with an initial ontology. We have designed the *OpenRobots Common Sense Ontology*, which precisely provides an upper set of concepts upon which the robot can add and connect new statements of the world.

This ontology is closely aligned on the open-source OPENCYC[5] upper ontology. OPENCYC proposes a large taxonomy of concepts and semantic relationships between concepts. We have been reusing OPENCYC identifiers and its taxonomy when possible (*i.e.*, when the concept we wanted to model did exist in OPENCYC), thus guaranteeing to a certain extent the alignement of our ontology with a major, standard, upper ontology. This potentially eases the exchange and addition of knowledge from other sources (the aforementionned OPENMIND project for instance, which belongs to the generic *LOD – Linked Open Data* concept, as commonly referred in the Semantic Web community. It includes querying Internet sources, but also exchanging knowledge with other robots).

In its current version, the *OpenRobots Common Sense Ontology* defines over 200 classes of concepts focused on concepts related to human environments. It includes both very broad categories like `SpatialThing`, `TemporalThing`, `Event` or `Action`, and much more concrete concepts as `Table`, `Book`, `black`... Robotic-specific concepts include `Robot` that is defined to be a kind of `IntelligentAgent`,

[5]http://www.opencyc.org

EmbodiedAgent and Artifact.

The ontology is stored as two OWL files (openrobots.oro.owl is the main one, while the second one holds the scenario-dependent instances, *i.e.* the model of the world). They are available online[6].

C. ORO main features

Besides simply storing and reasoning about knowledge, we have implemented in ORO several features that we claim useful for human-robot interaction: events registration, categorization capabilities, independent cognitive models for each agent the robot knows and different profiles of memory.

1) The events framework: ORO allows external modules to be triggered when specific events occur. Several type of event can be registered (*e.g.*, *a new instance of a given class appears* or *a set of facts becomes true*). For instance, an expressions like: *"Tell me when any kind of tableware appears on the table."* can be translated into the event $Evt(obj) \Leftrightarrow \exists obj / type(obj, Tableware) \wedge isOn(obj, table)$.

This has been designed to enable the implementation of reactive behaviours ("waking up modules") that would take advantage of the inference capabilities of the reasoner.

2) Categorization: We have implemented several algorithms (common ancestors, computation of the best discriminant) to help the robot to cluster a set of concepts based on their symbolic similarities (common properties, common ancestors). The *Spy game* scenario (section V-C) shows a usage of these categorization abilities.

3) Modelling of alternative cognitive model: As shown in Figure 1, ORO can store independent cognitive models for each agent it interacts with. When ORO actually identifies a new agent (or infers that some object is an agent), it creates a new, separate, RDF triple storage. External modules like supervision or dedicated *perspective taking* components may then store facts or beliefs about agents' beliefs [9]. This allows to store and reason on two different (and possibly inconsistent) models of the world.

4) Memory profiles: We have designed a simplified bio-inspired memory model that allows us to store statements in different *memory profiles*. These include *short term memory* and *long term memory*. Each profile is characterized with a lifetime, which is assigned to the stored facts. When the lifetime of a fact expires, ORO automatically removes it.

D. The semantic level

ORO, by claiming its "cognitive" nature, brings something more than a simple knowledge storage (*i.e.* what we would expect from a database for instance).

As an ontology-based knowledge processing tool, ORO allows us to *connect* together pieces of knowledge in a coherent way, that is, to put chuncks of information about the world in a symbolic *context*.

This opens many new opportunities in the design of robotics architecture, not only providing individual modules (even low-level ones, like perception) with advanced reasoning abilities, but also by aggregation of knowledge: this semantic level allows to cleanly put together sources of information that are traditionally difficult to combine, like visual perception, geometrical reasoning, common-sense knowledge or human input.

An unexpected example of this "semantic bonus" emerged while we were setting up the "Odd One Out" experiment. The perception routines provided segmented blobs corresponding to objects, along with their colours. The supervision would then feed ORO with the visible objects. At some point, ORO suddently refused to add an object. What seemed at first as a communication bug between modules, was actually the consequence of a consistency check by ORO: Because of bad light conditions, the color recognition was not very reliable, and the same object was set to have two different colours at the same time. That was inferred as impossible by ORO and thus discarded. While we didn't had time to implement it, this kind of failure can clearly be used to improve low-level perception results by "closing the loop" with high-level, symbolic knowledge.

IV. Integration into robotic architectures

In the previous section we had an overview of the active processes that are embedded in ORO and we explained how ORO is a common knowledge framework for other modules to represent and share, in a consistent way, the knowledge they produce. But ORO integrates only few of the cognitive abilities required to build a complete robot.

In [10], Langley et al. propose a list of capabilities that a cognitive architecture should support to *"cover the full range of human-level intelligent activities"*. These criteria are well suited to show how ORO is integrated with components that are currently implemented in the different robots in which ORO has been deployed.

A. Perception, recognition and categorization

Recognition of objects or situations requires the identification of unique and invariant patterns for each objects. To this end, robots must percieve their environment either from raw sensors like cameras or laser scanners or through higher level mecanisms like motion capture or tags.

However, ORO is not directly fed by the sensors, but by abstracted intermediate geometric models. MOVE3D [11] is such an intermediate world representation. Not only these module feed ORO about object visibility, but also compute *spatial relationships* between objects.

B. Decision making, planning, execution

Decision making (commonly attributed to the *supervision* module) and *planning* use symbolic representation and reasoning systems to retrieve the current beliefs of the robot or other agents (which include the current believed state of the world) and to state goals, actions and plans.

[6] http://www.laas.fr/~slemaign/oro-server/oro-ontology.html

Supervisors, like SHARY [12] or CRAM[7] [14] use ORO as well for categorization (see the *Odd One Out* and *Spy game* example below) or reactive supervision through the event system.

C. Interaction and communication

One of the first expectation of a framework like ORO is to ease interaction with other agents (both humans and robots) thanks to higher level symbolic representation of objects. ORO has already been connected to several human-robot interfaces (like speech-recognition and synthesis system[8]), and on-going developments target the integration of natural language processing and bindings with external resources like WORDNET.

ORO is also used in experiments related to *Perspective Taking* (see the *Spy game* example below) where the ability of the robot to model human's perspective on the world is a prerequisite for the interaction.

D. Monitoring, remembering, reflection and learning

ORO eases the implemention of some of these higher cognitive functions (referred by Sloman as *"metamanagement mechanisms"*).

An example of *monitoring* has already been given previously with the perception of color mismatch.

In its current state, ORO already has a (naive) model of memory as described in Section III-C4.

Reflection (*i.e.*, the ability for the robot to think and talk about its own knowledge) is an immediate consequence of the explicit and uniform modelling of the knowledge and its structure in an ontology.

Finally, while *learning* is not yet tackled *per se* in ORO, in our experiments we show humans teaching the robot new symbolic structures (*Odd One Out* scenario).

E. Technical aspects of ORO integration

ORO was designed to be portable (command-line application written in pure Java) and easy to integrate in existing robotic cognitive architecture by having few dependencies (besides the Java VM, the only two dependencies are Jena, the RDF triple store, and Pellet, the reasoner).

ORO uses a custom (very simple) ASCII protocol over TCP sockets that guarantees almost universal compatibility, and easy testing and debugging with standard tools like Telnet.

Several middleware bindings and language-specific wrappers have been developed to ease the integration of ORO in existing software. Most notably, ORO plays nicely with the ROS[9] and YARP[10] middlewares, and C++ (`liboro`) and Python (`pyoro`) have well maintained wrappers. Bindings for TCL are also available.

[7] CRAM (Cognitive Robotic Abstract Machine) is a RPL-derived [13] framework for rapid development of cognitive robot control programs we currently develop.

[8] The CSLU Toolkit, http://cslu.cse.ogi.edu/toolkit/

[9] The *Robotic Operating System*, http://www.ros.org/

[10] http://eris.liralab.it/yarp/

V. EXPERIMENTAL USAGES

ORO has already been deployed on three different robots:

- the *BERT2* robot at BRL (YARP-based architecture)
- the *Kimp* robot at TUM-IAS (ROS-based architecture),
- the *Jido* robot at LAAS-CNRS (based on the in-house Pocolibs middleware and the C++ `liboro` wrapper)

This is to illustrate its effective integration and sketch potentialities of its use.

A. Knowledge acquisition: Point & Learn

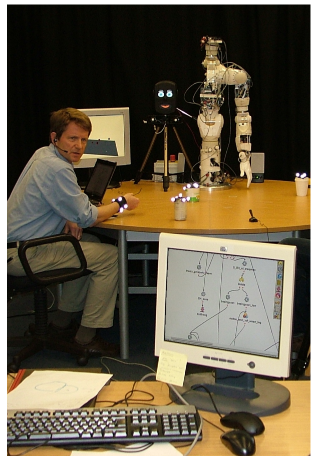


Fig. 2. Teaching the Bert robot new objects

We have implemented a *Point & learn* behaviour on the Bert robot[11] (Figure 2): a human shows an object to the robot, and if the robot sees it for the first time, it will ask for the name and the type of the object.

The object perception module relies on motion capture to identify and localize objects. A *detection* module was responsible for updating ORO with the list of objects currently seen by the robot as well as their state (moving or not) and their relations to other objects (touching or not). On the other end, a human-robot interface based on the CLSU Toolkit was in charge of speech recognition, speech synthesis and natural language analysis.

By querying ORO for moving objects, the human-robot interface retrieves the object ID that had the focus of attention (last moving object), and asks the human for a name and a type if the object was new.

Figure 3 reproduces a typical dialog with Bert.

At the end of this sequence, two more RDF statements are added to the robot knowledge base: `[5001 rdfs:label "coffee-cup"]` and `[5001 rdf:type Cup]`.

Due to the limitation of the speech recognition software, only a predefined set of names or type could be recognized, thus constraining the learning skills of the experiment.

B. Odd One Out

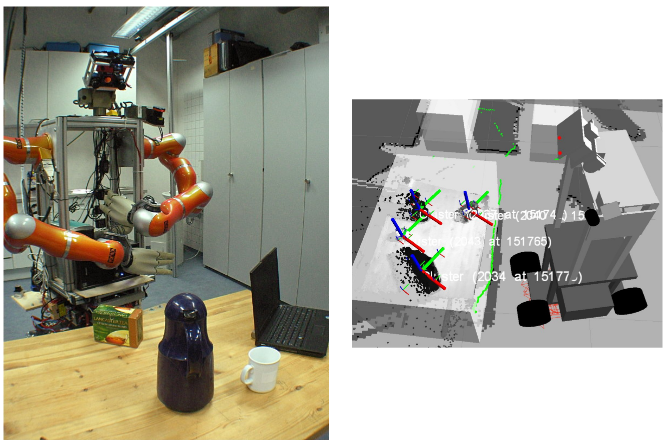
The *Odd One Out* scenario extends the *Point & Learn* experiment and completes an on-going experiment at the IAS laboratory where a robot is asked to list missing items on a table being set, based on probabilistic reasoning on previously recorded observations [15].

We use ORO to introduce human interactions and commonsense reasoning: the robot picks one after the other objects (tea box, IceTea bottle, mug...) placed on the table that it doesn't know yet, it shows them to the human and asks about the names and type of each of the new objects until the human provides the name of a concept the robot already knows

[11] This experiment was conducted in the frame of the European CHRIS project

bert	Initializing... *[about 5 sec]* ...What's next?
human	*[moves an object]*
bert	*[does not know the object]* How is it called?
human	coffee-cup
bert	Did you say coffee-cup?
human	yes
bert	Ok. Now I know. What kind of object is coffee-cup?
human	a cup
bert	Did you say cup?
human	yes
bert	So coffee cup is a cup. What's next?

Fig. 3. Transcript of a chat with the Bert robot



(a) (b)

Fig. 4. (a) Kimp, looking for objects it may know, and (b) viewed in RViz. The clusters of point are given an unique identifier by the perception that allow the supervision create the link between the physical objects and their symbolic representation in ORO.

(Figure V-B). The robot then tells which objects do not belong to a typical breakfast table (objects that are neither food or tableware).

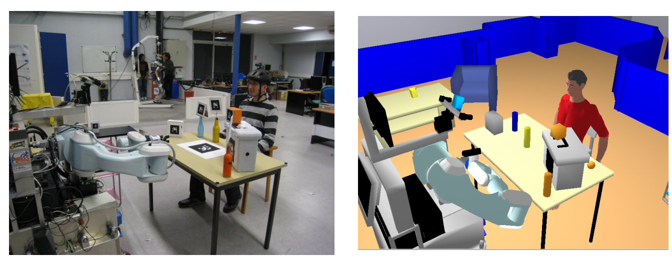
The supervision (CRAM) automatically maintains up-to-date the ORO server whenever objects enter or leave the field of view. Therefore, integration of ORO can be seen as an extension to the robot's belief state that not only contains abstract identifiers of the internal object representation used in plans, but also the semantics and roles of objects in the scenario.

The human interacts with the robot through a dedicated XMPP bridge, allowing to *chat* with the robot with a standard Jabber instant messaging client. Figure 5 corresponds to a chat session with Kimp.

By asking in loop the human for the categories of an object until it can connect it to a concept it already knows, the robot accurately anchors a perception in its symbolic model and it is able to reason about it. At the end of the experiment, the robot identifies and returns the odd objects for the breakfast table (*i.e.*, in our example, objects that are neither `Tableware` or `Food`).

kimp	Servus! The bavarian robot is up and fit!
kimp	What is the name of the WHITE CLUSTER on the TABLE?
human	"Kimp's mug" *[Kimp adds* `OBJ001 rdfs:label "Kimp's mug"` *stmt]*
kimp	What kind of object is Kimp's mug?
human	Mug
kimp	What is a Mug? *[Kimp adds* `OBJ001 rdf:type Mug` *but it doesn't know the* Mug *concept]*
human	Tableware *[Kimp knows about* Tableware *- belongs to the common-sense ontology. It adds* `Mug rdfs:subClassOf Tableware`*]*
	(...)
kimp	What is the name of the BLACK CLUSTER on the TABLE?
human	my laptop
kimp	What kind of object is my laptop?
human	Computer
kimp	What is a Computer?
human	Artifact
kimp	Do you want me to learn more objects?
human	no
kimp	These objects are odd: my laptop

Fig. 5. Transcript of a Jabber session with the robot Kimp. Compared to dialog with Bert (3), we see here that the robot anchors the new objects in its already acquired knowledge.



(a) (b)

Fig. 6. Spy game scenario: (a) Real environment and (b) 3D environment model, viewed in MOVE3D.

C. The Spy game

This game is based on the traditional children game "I Spy". The idea is to discover the object or concept one of the participants is thinking of by asking questions such as: "Is it green? Is it a machine? Is it on your left?", etc. When playing, children exploit their knowledge about the world while categorizing and describing objects through useful discriminants that will allow them to find out the answer as fast as possible [16].

The scenario for this game (Figure V-C) consists on a face-to-face interaction where the human thinks of an object present in the environment, while the robot queries the human until either discovering the object or giving up, if no object was found. A categorization example is presented in Figure 7. The game starts with the human user giving a first hint (communication is done through a keyboard and screen), allowing the robot to start the search filtering those objects that fulfill this first description. Based on this subset, ORO provides a descriptor (or set of descriptors) that allows a maximum discrimination among objects in the subset. The robot queries the user about the value of the descriptor (or the most discriminant among the set of descriptors) and with this new information, the current subset of objects is filtered again. The process is repeated until either obtaining a single object that fulfills all the descriptor values, or failing (*i.e.* no object found).

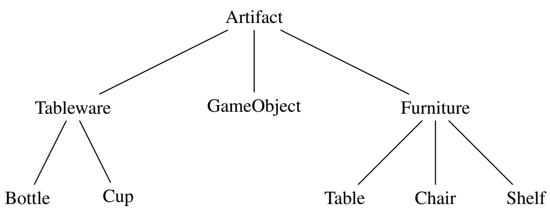


Fig. 7. Example of object categorization used in the scenario.

We have integrated the game in the LAAS-CNRS Jido robot [17]. Objects are identified through a tag-based vision approach[12] and motion capture is used for human tracking. Their descriptions regarding categories (type of object) and features (color, shape) are manually given in advance. Spatial relationships (front, back, left, etc, and in, on and next to) and visibility (only visible objects for both agents can be considered in the game) are automatically computed on-line [9]. Figure 8 shows an example of a round game.

human	It is a tableware.
jido	*[retrieves possible objects: blue-bottle, yellow-bottle, orange-bottle, cup-with-handle]* *[removes non visible objects: blue-bottle, yellow-bottle, cup-with-handle]* *[obtains discriminants: type, color.]* Which type of object is: bottle or cup?
human	Bottle.
jido	*[obtains possible objects: blue-bottle, yellow-bottle.]* *[obtains discriminants: color.]* What color the object is: blue or yellow?
human	Blue.
jido	*[obtains possible objects: blue-bottle.]* The object is the blue-bottle!

Fig. 8. Example of the robot playing Spy game.

VI. Conclusion

In this paper we have presented ORO, a knowledge processing module for cognitive robotics. We also briefly introduced the *OpenRobots Common Sense Ontology*. ORO is a socket server aimed to be run on robots that firstly maintains a consistent storage of facts, represented as RDF triples, and secondarily runs several background processes, including ontology classification and reasoning, management of several independent models for each different agent the robot meets, and updating of statements according to bio-inspired memory models.

While ORO is already useful and has been actually used in several human-robot interaction scenarii on three different robotic plateform, one of our aims is to offer a complete cognitive library for practical use in semantic-rich environments and human-robot interaction situations.

Several areas of improvement are currently being investigated: the system currently acquires knowledge from its perception and own modelling of plans and motivation. At simple levels, the robot can as well acquire knowledge from its verbal interaction with humans. We would like to extend these sources with a more generic access to external resources (including on-line resources like Wikipedia). This will require to improve the current natural language processing capabilities.

Other areas of research include richer models of memory (including reinforcement learning), handling of inconsistent states of the knowledge base (explanation of inconsistencies, solution to pro-actively solve them), implementation of mechanisms to pro-actively look for new relations between concepts (*curiosity* module) and the design of a generic framework for acquisition and filtering of knowledge that could be used both in human-robot verbal interaction and when retrieving facts from the Internet.

[12] ARToolKit: http://www.hitl.washington.edu/artoolkit/

Acknowledgements

Part of this work has been funded by the European CHRIS FP7 project (`http://www.chrisfp7.eu/`). We thank hereby INSERM and BRL teams for their support.

References

[1] A. Leslie, "Theory of mind'as a mechanism of selective attention," *The new cognitive neurosciences*, pp. 1235–1247, 2000.

[2] J. McCarthy, "From here to human-level AI," *Artificial Intelligence*, 2007.

[3] A. Sloman, J. Wyatt, and N. Hawes, "Long term requirements for cognitive robotics," in *In Proc. Cognitive Robotics O6 Workshop, AAAI'06. AAAI*, AAAI Press, 2006.

[4] H. Levesque and G. Lakemeyer, "Cognitive robotics," in *Chapter 24 in Handbook of Knowledge Representation*, Elsevier, 2007.

[5] J. Gunderson and L. Gunderson, *Robots, Reasoning, and Reification*. Springer, 2008.

[6] M. Daoutis, S. Coradeshia, and A. Loutfia, "Grounding commonsense knowledge in intelligent systems," *Journal of Ambient Intelligence and Smart Environments*, pp. 311–321, 2009.

[7] M. Tenorth and M. Beetz, "Knowrob - knowledge processing for autonomous personal robots," in *IEEE/RSJ International Conference on Intelligent RObots and Systems.*, 2009.

[8] P. Singh, T. Lin, E. Mueller, G. Lim, T. Perkins, and W. Li Zhu, "Open mind common sense: Knowledge acquisition from the general public," *Lecture Notes in Computer Science*, pp. 1223–1237, 2002.

[9] R. Ros, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Solving ambiguities with perspective taking," in *5th ACM/IEEE International Conference on Human-Robot Interaction*, 2010. To Appear.

[10] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cognitive Systems Research*, vol. 10, pp. 141–160, June 2009.

[11] L. Marin, E. A. Sisbot, and R. Alami, "Geometric tools for perspective taking for human-robot interaction," in *Mexican International Conference on Artificial Intelligence (MICAI 2008)*, (Mexico City, Mexico), October 2008.

[12] A. Clodic, H. Cao, S. Alili, V. Montreuil, R. Alami, and R. Chatila, "SHARY: a supervision system adapted to human-robot interaction," in *11th International Symposium on Experimental Robotics*, 2008.

[13] D. McDermott, "A Reactive Plan Language," research report yaleu/dcs/rr-864, Yale University, 1991.

[14] M. Beetz, L. Mösenlechner, and M. Tenorth, "CRAM — A Cognitive Robot Abstract Machine for everyday manipulation in human environments," in *International Conference on Intelligent RObots and Systems*, 2010. Submitted.

[15] D. Pangercic, R. Tavcar, M. Tenorth, and M. Beetz, "Visual scene detection and interpretation using encyclopedic knowledge and formal description logic," in *Proceedings of the International Conference on Advanced Robotics (ICAR).*, 2009.

[16] H. Moll and M. Tomasello, "Level 1 perspective-taking at 24 months of age," *British Journal of Developmental Psychology*, vol. 24, no. 3, pp. 603–614, 2006.

[17] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *The International Journal of Robotics Research*, vol. 17, no. 4, p. 315, 1998.