The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

ORO, a knowledge management platform 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 embeddable knowledge processing framework, 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 reasoning 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 to deal with a lot of symbolic 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, a rich symbolic model of the environment and rules that will allow to reason on this knowledge are needed.

Moreover, if the robot directly interacts with humans, it may even be necessary to take the human perspective in order to perceive and model the human own beliefs of 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 his/her view). If the human asks for "the jam", the robot must infer that he/she is referring to the jam he/she sees. This ability to think about other agents' mental states is part of the so-called *theory of mind*. Humans rely on such capabilities to develop social interactions amongst them. Thus we believe that robots should be provided with these abilities in order to achieve and maintain an effective interaction with humans.

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 expressiveness while remaining well suited for machine processing in order to be used by the robot.

This paper introduces ORO, an easy-to-deploy platform for symbolic knowledge storage and reasoning. Based on an ontology, it brings several advanced cognitive features to robotic architectures such as categorization or explicit modeling of agents mental states, along with a common-sense ontology, focused on human-robot interaction needs.

As an ontology-based knowledge processing tool, ORO allows to *connect* together pieces of knowledge in a coherent way, that is, to put chunks 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 on symbolic facts they could produce, but also by aggregation of knowledge: the semantic layer we introduce 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.

The novelty of this work lies in the fast, standard-based knowledge store we have developed, in the library of high-level, built-in cognitive functions that are available to the robot software designers, and in the global approach of knowledge processing we propose within the robot.

We demonstrate this tool on three different robotic architectures and in experiments that involve grounded, symbolic interaction and decision-making in human environments.

The paper is organized the following way: we present a brief overview of the current cognitive and knowledge processing approaches within the robotics community in Section II. The ORO knowledge processing platform is introduced in Section III and concrete applications on three different robotic architectures are described in Section IV. Section V concludes the paper with some perspectives.

II. Related work

Pioneering works on questions related to cognition in robotics include papers by McCarthy [1], Sloman et al. [2] or Levesque and Lakemeyer [3]. Most of the challenges of cognitive robotics can be summarized from these articles.

In the field of symbolic knowledge processing for robots Gunderson and Gunderson [4] introduce the concept of reification (based on both recognition and pre-afference) as an

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

intermediate step between pattern recognition and symbol grounding. Their underlying storage of knowledge relies on ontologies and bio-inspired memory model.

Suh et al. [5] introduce OMRKF, an ontology-based reasoning framework. They try to tackle the grounding issue by storing low-level facts (like SIFT visual features) in a layered symbolic architecture that works well in simple sensori-motor spaces, but this approach raises concerns regarding scalability and management of more complex entities or interactions.

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] develop 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 then, for instance, more difficult to detect.

III. ORO, A ONTOLOGY-BASED KNOWLEDGE PROCESSOR

A. Architecture

The open-source ORO platform[1] is designed as a service: it primarily works as an intelligent blackboard that allows other modules in the robot to push or pull asserted and inferred knowledge to a central repository. It is built around a socket-based server, built itself on the top of a standard RDF triples store. Figure 1 illustrates the overall architecture. The *front-end* accepts and manages connections from client components. The clients' requests are processed by a set of internal *modules*. Besides basic operations like knowledge retrieval and storage, *plugins* can be loaded to add advanced cognitive and human-robot interaction abilities (see below). Ultimately, the modules rely on several parallel ontology *backends*. The knowledge is actually stored in these backends.

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 Description Logic[2], which is the decidable part of OWL. The underlying RDF triples storage is the Jena framework[3]. We use it in conjunction with the Pellet[4] reasoner to ensure the continuous classification of the storage. During run-time, statements are permanently added, removed or queried to and from ORO by the robot's components (perception, supervision, planner, etc.). ORO is responsible for continuously maintaining a classified, up-to-date set of statements.

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.

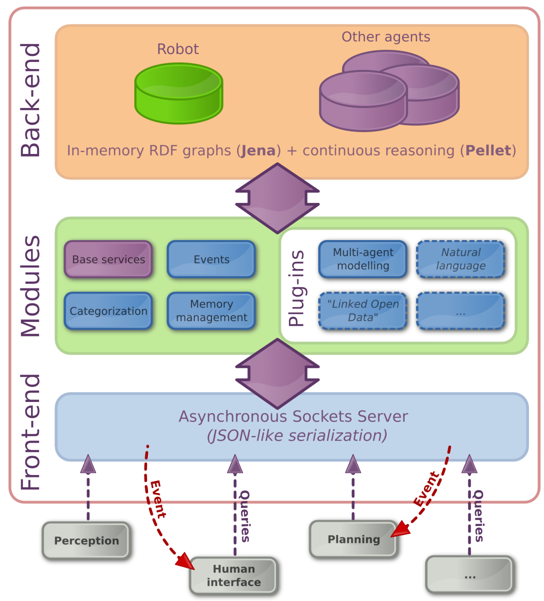


Fig. 1. Overview of the ORO architecture

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*, both declarative –*snow is cold*– and procedural –*how to open a door*.

Several significant projects are trying to provide machine-processable repository of common sense facts produced by humans (the OPENMIND project[5], for instance). These knowledge bases are valuable but remain difficult to use because of both their incompleteness and the lack of good connections with underlying, unambiguous concepts.

The knowledge that the robot acquires (by perception or interaction) needs however to be interconnected with what the robot already knows, *i.e.* anchored, to become actually useful. This requires at least an agreement on common identifiers to symbolize identical concepts. ORO can be loaded with an

[1] Project homepage: http://homepages.laas.fr/slemaign/oro-server
[2] http://www.w3.org/TR/owl2-overview/
[3] http://jena.sourceforge.net
[4] http://clarkparsia.com/pellet/
[5] http://www.openmind.org/

initial set of statements and we have designed the *OpenRobots Common Sense Ontology*[6] that defines a restraint set of concepts and accordingly, a vocabulary that can be used by all the modules of the robot to unambiguously add or query facts. The same ontology permits as well to assert rules and logical properties in a declarative way, used for inference.

The *OpenRobots Common Sense Ontology* is closely aligned on the open-source OPENCYC[7] 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, thus guaranteeing the alignment of our ontology with a major, standard, upper ontology. This potentially eases the exchange and addition of knowledge from other sources (by querying online resources like the aforementioned OPENMIND project or WORDNET, the ontology of English language) but also, by exchanging knowledge with other robots.

The *OpenRobots Common Sense Ontology* defines classes (56 are currently defined) and predicates (60 are currently defined) focused on concepts useful for interaction with humans. It includes both very broad categories like `SpatialThing`, `Event` or `Action`, and much more concrete concepts as `Table`, `Book` or the colour `blue`. Predicates allow to describe the state of the agents and the world with relations like `isOn`, `sees`, `currentlyPerforms`. Robotic-specific concepts include `Robot`, which is defined to be a kind of `IntelligentAgent`, `EmbodiedAgent` and `Artifact`.

C. ORO features

Besides simply storing and reasoning about knowledge, we have implemented in ORO several functions to manage knowledge at higher level: events registration, independent cognitive models for each agent the robot knows, categorization capabilities and different profiles of memory.

1) Base functionalities: As expected from any knowledge base, ORO offers an extended set of methods for standard processing of facts. It includes:

- inserting facts (*i.e.*, RDF triples), removing them, updating them,
- removing statements based on patterns,
- consistency checking, adding statements with consistency constraint (only if the new fact does not lead to inconsistencies),
- looking up for concepts, with multi-lingual support,
- querying the ontology with a combination of patterns (for instance `* isOn table`) and filters (for instance `weight < 150.0`),
- executing standard SPARQL[8] queries.

2) The events framework: ORO allows external modules to be triggered when specific events occur. For instance, when a logical sentence becomes true or false, or if a new instance of a certain class is added. This proves particularly useful for reactive supervision: in one of our experiment, the supervisor registers at start-up an event that is triggered by new statements of kind: `?agent desires ?situation`. When a user says "I want you to take to bottle", the sentence is translated into a set of statement (`human desires SIT_u87fs, SIT_u87fs rdf:type Take`, etc.) that triggers the supervisor as soon as they are added to ORO.

The event framework also takes advantage of the inference capabilities of ORO. Thus an event can be indirectly triggered if its triggering conditions can be inferred to be true.

3) Representation of alternative cognitive models: As shown in Figure 1, ORO stores independent cognitive models for each agent it interacts with. When ORO actually identifies a new agent (or infers that some instance is an agent), it automatically creates a new, separate, RDF triple storage. External modules like supervision or dedicated *perspective taking* components [8] may then store facts or beliefs about the agents' beliefs. This allows to store and reason on different (and possibly globally inconsistent) models of the world.

4) Categorization: We have implemented several algorithms (common ancestors, computation of the best discriminant, see [9]) to help the robot cluster a set of concepts based on their symbolic similarities (common properties, common ancestors). One particular application of these functions is discrimination: when interacting with a user, the robot often needs to proceed to concept disambiguation. For instance, a user may refer to a "Bottle" whereas two bottles are currently visible: the discrimination routines can identify possible (symbolic) differences (maybe the colour or the size of the bottles) that permit the robot to ask an accurate question to the user. This discrimination can occur from the robot perspective or from a specific agent perspective. The *Spy game* scenario (section IV-D) shows an example of these categorization abilities.

5) 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.

IV. EXPERIMENTAL USAGES

To illustrate the effective integration and sketch potentialities of ORO we next describe three different tasks that have been conducted on three completely different platforms:

- the *BERT2* robot at BRL (YARP-based architecture)
- the *Rosie* robot at TUM-IAS (ROS-based architecture),
- the *Jido* robot at LAAS-CNRS (based on the LAAS Pocolibs middleware)

A. Technical background of ORO integration

ORO was designed to be portable (command-line application written in pure Java) and easy to integrate in existing

[6]http://homepages.laas.fr/slemaign/oro-server/oro-ontology.html
[7]http://www.opencyc.org
[8]http://www.w3.org/TR/rdf-sparql-query/

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.

B. Knowledge acquisition: Point & Learn

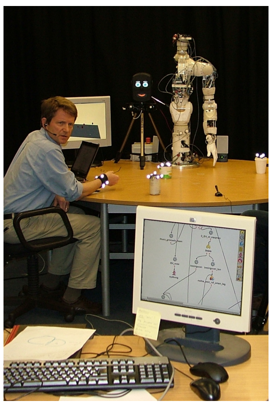


Fig. 2. Teaching the Bert robot new objects

We have implemented a *Point & learn* behaviour on the Bert robot [10] (Figure 2): the user shows an object to the robot, and if the robot sees it for the first time, it will ask for its name and type.

The object perception module relies on motion capture (VICON system) to identify and localize objects. A so-called *primitive detection* module is 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[11] is in charge of speech recognition, speech synthesis and basic natural language processing.

By querying ORO for moving objects, the interface retrieves the object ID that has the focus of attention (last moving object), and asks the human for a name and a type if the object is 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]`.

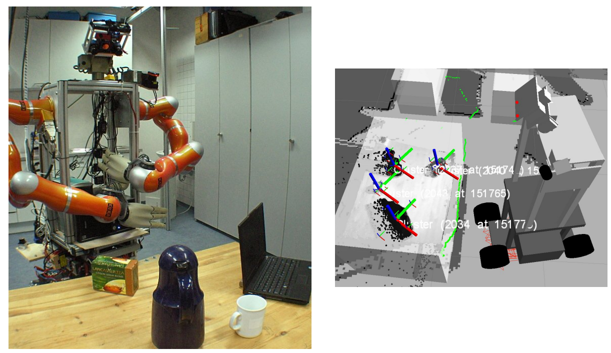
Due to the limitation of the speech recognition software, only a predefined set of names or types could be recognized, thus preventing the robot to add completely original objects.

C. 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.

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) Rosie, 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.

We use ORO to introduce human interactions and common-sense reasoning: the robot picks an unknown object from the table, shows it to the user, and asks about its name and type (Figure 4). The user continues to describe the object (through concepts) until a concept known by the robot is given. The learning process starts over again with another unknown object. Once all objects are learned, the robot tells which objects do not belong to a typical breakfast table (*i.e.* objects that are neither food or tableware). The human interacts with the robot through a dedicated XMPP bridge, allowing to chat with the robot with a standard Jabber messaging client. Figure 5 corresponds to a chat session with Rosie.

The supervision (CRAM[12] [11]) automatically updates the ORO server whenever an object enters or leaves the field of view. Therefore, the 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.

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

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

[11] http://cslu.cse.ogi.edu/toolkit/

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

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

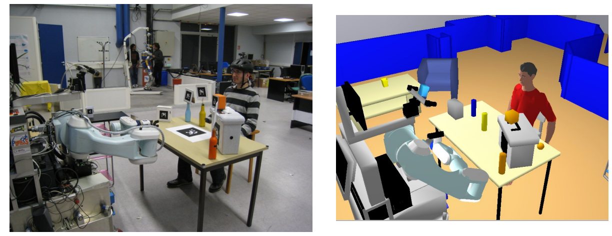
Fig. 5. Transcript of a Jabber session with the robot Rosie. Compared to dialog with Bert (3), we see here that the robot anchors the new objects in its already acquired knowledge.

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 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`).

An unexpected example of what the symbolic reasoning layer brings to more traditional robotic architectures emerged during 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 suddenly refused to add an object. What seemed at first 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. This kind of logical failure can be used to improve low-level perception results by “closing the loop” with high-level, symbolic knowledge.

D. 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 allow them to find out the answer as fast as possible while including perspective taking abilities [12].



(a) (b)

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

The scenario for this game (Figure 6) 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).

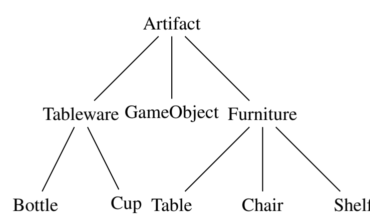


Fig. 7. Example of object categorization used in the scenario.

We have integrated the game in the LAAS-CNRS Jido robot [13]. Objects are identified through a tag-based vision approach[13] 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 by the MHP/MOVE3D geometric reasoner and planner [14]. Figure 8 shows an example of a round game.

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

human	It is a tableware.
jido	*[retrieves possible objects: blue-bottle, yellow-bottle, orange-bottle, cup-with-handle]*
	[keeps 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.

V. 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 *(1)* maintains a consistent storage of facts, represented as RDF triples, and *(2)* 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.

Figure 9 presents a synthetic evaluation of ORO performances on standard knowledge manipulation operations. This shows that ORO is fast enough for on-line usage on a robot. A thorough comparison of ORO performances with other reasoning frameworks remains to be conducted.

While ORO has been already used in several human-robot interaction scenarii on three different robotic platforms, one of our aims is to offer a comprehensive cognitive library for practical use in semantic-rich environments and human-robot interaction situations. Several areas of improvement are currently being investigated: more generic access to external resources (including on-line resources like Wikipedia or WORDNET), integration with natural language processing capabilities to add facts or query the ontology from verbal interaction with users, generic management of alternate "views on the world" (*MicroTheories* in OPENCYC terminology), and a richer representation of time constraints and plans.

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.

	duration in ms	stmts/sec
Add 10000 stmts + classification	1380	7245
Consistency check	< 1	
Query 1 (inheritance)	293	34130
Query 2 (logical properties)	135	73855
Query 3 (conjunction)	1342	7447

Fig. 9. Performance evaluation of ORO server: insertion of a set of 10000 statements to a pre-loaded ontology (`testsuite.oro.owl`, expressiveness: $SHOIQ(D)$, reasoner: Pellet 2.0.2), and retrieval through three different kind of inferences. Results are averaged on 10 tries.

ACKNOWLEDGMENTS

Part of this work has been conducted within the EU CHRIS project (`http://www.chrisfp7.eu/`) funded by the E.C. Division FP7-IST under Contract 215805. We also acknowledge support of Agence Nationale de la Recherche ANR (AMORCES/PSIROB project) and part of this work has been supported by a Marie Curie Intra European Fellowship.

REFERENCES

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

[2] 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.

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

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

[5] I. Suh, G. Lim, W. Hwang, H. Suh, J. Choi, and Y. Park, "Ontology-based multi-layered robot knowledge framework (omrkf) for robot intelligence," *Proceeding of the IEEE the IROS*, 2007.

[6] M. Daoutis, S. Coradeshi, and A. Loutfi, "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] 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.

[9] R. Ros, S. Lemaignan, E. A. Sisbot, R. Alami, J. Steinwender, K. Hamann, and F. Warneken, "Which one? grounding the referent based on efficient human-robot interaction," in *19th IEEE International Symposium in Robot and Human Interactive Communication*, 2010. To appear.

[10] S. Lallée, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, T. van Der Zant, F. Warneken, and P. F. Dominey, "Towards a platform-independent cooperative human-robot interaction system: I. perception," in *International Conference on Intelligent RObots and Systems*, 2010. To appear.

[11] 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.

[12] 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.

[13] 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.

[14] 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.