OBEliSK: Novel Knowledgebase of Object Features and Exchange Strategies

David Cabañeros Blanco, Ana Belén Rodríguez Arias, Víctor Fernández-Carbajales Cañete, and Joaquín Canseco Suárez

TREELOGIC Telemática y Lógica Racional para la Empresa Europea
Parque Tecnológico de Asturias - Parcela 30. E33428 - Llanera, Asturias - SPAIN
{`david.cabaneros, ana.rodriguez, victor.fernandez,`
`joaquin.canseco`}`@treelogic.com`

Abstract. This paper presents the design and development of a system intended for storing, querying and managing all required data related to a human-robot exchange process. This information ranges from object attributes involved in visual object recognition to grasping and delivering features for scenario-focused object exchanges. Our system acts as a bridge between visual perception and control systems in a robotic setup intended to collaborate with human partners. The perception module provides information about the exchange environment, such as object identification and its current pose, provisioning the robotic control system with the required knowledge for executing the actions involved in a fluent human-robot object exchange. In order to achieve these goals, a semantic-ontological approach has been selected favouring system's interoperability and extensibility, complemented with a set of utilities developed ad-hoc for easing the knowledge inference, query and management. As a result, the developed knowledgebase provides a completeness level not previously reached in related state of art approaches.

Keywords: Ontologies, Knowledge Representation, Handling Affordances, Semantic Modelling, Assistive Robotics

1 Introduction

The work described in this paper is the design and development of a knowledgebase about the domain elements involved in the action of exchanging common objects between humans and robotic agents. Our approach requires an in-depth study of the state of the art, inputs from the perception system and a clear definition of the outputs required by the robotic control architecture.

For this purpose, the initial specification of the problem has been divided into two main branches, according to the knowledge modelling needs:

- The *object model* comprises its attributes, such as name, 3-D model, views, geometrical properties and related exchange strategies.
- The *exchange model* is based on the actions required for the exchange process as a whole. This process is composed by a set of attributes which describes

an exchange strategy. Each strategy is composed by specific grasp and handover methods that can be combined in several ways, providing different behavioural options to the robotic system's control layer based on the perceived situation during the object manipulation phase.

This data-representation model has been designed taking into account future scalability issues, in order to provide not only a complete knowledgebase, but also an extensible and adaptable database of objects and their exchange process. The adopted hierarchical classification of the knowledgebase model makes possible to upgrade the existing set of model entities, such as objects, grasps and deliveries, according to new challenges in the robotics field.

The main purpose of this knowledgebase is to model and transfer the acquired knowledge from human-human exchange experiments to a robotic system in order to achieve a fluent exchange between human and robotic agents.

The remainder of this paper is organized as follows. The next section (Section 2) introduces the theoretical concepts involved in our work. In Section 3, an in-depth state of the art analysis is performed. Section 4 describes in detail the design and development process of OBELiSK (Object Exchange applied Semantic Knowledgebase). In Section 5 we perform a comparison between our results and several alternatives studied in the state of the art. Additionally, the knowledgebase management tool is also introduced. Finally, we present our conclusions in Section 6.

2 Theoretical frame

Specific-domain knowledge is defined as the information about a particular field that can be used as approach to solve problems defined in that domain [1]. Ontological representations are intended to model a formal representation of the knowledge about some specific domain by means of concepts and relations between them as basic building blocks [2]. These systems are described by a computational model, composed by a set of entities corresponding to real world items, such as agents, objects or events connected by domain-specific rules. The final purpose of this representation is to retrieve understanding detached from the experience, represented as statements about the domain and stored into a computational model. In this way, applications may use this model to perform reasoning and get answers to questions about domain entities and the way in which they are related.

The semantic-ontological model representation previously introduced requires an approach far from the "classical" relational database [3], so in this case we considered graph databases [4]. The main advantage, among others [5], of such representation is the flexibility that provides when linking related entities, attributes and properties. This allows to make the most of the data, while reduces the inherent complexity of traditional databases. For representing data within this approach while extending its usefulness, the Semantic Web [6], intended to provide a common framework for sharing and reusing data across application, enterprise and community boundaries, defines a standardized set of technologies, arranged in a hierarchical architecture where each layer relies on the layers below. This framework, known as RDF (Resource Description Framework) [7], was designed as a method for splitting knowledge into small pieces of data, complemented with a set of rules defining the semantics of these individual fragments of isolated information. RDF/XML format is more intended for machine-reading than for human-reading, so the following formats were defined in order to ease the reading of RDF documents for humans:

- *Notation3* is a compact alternative to RDF/XML syntax, focused on human readability. It also extends RDF/XML to allow more expressiveness by means of inclusion of RDF rules.
- *Turtle* stands for a particular syntax defined as a subset of the *Notation3* language. As opposed to RDF/XML, Turtle does not depend on XML. Although it is not recognized as standard, Turtle is widely adopted for writing and parsing RDF data.
- *N-Triples* is as a subset of the Turtle format. Its goal is to serialize RDF data into plain text in order to simplify its transmission.

Also, in order to represent the envisioned model, the Web Ontology Language (OWL), was selected [8]. A key benefit of the semantic-ontological approach is its reusability empowering, encouraging the creation of new resources based on the existing ones. On the one hand, the designed knowledge representations might be re-used in the development of different systems addressing similar purposes. On the other hand, this scheme brings interoperatibility between heterogeneous systems according to a consensuated knowledge representation. Companies and organizations seeking to encourage this kind of knowledge reusing techniques are becoming more numerous.

As an approach for improving the expressiveness of traditional propositional logic, Description Logic (DL) languages were introduced as knowledge representation methods, applied mainly in the fields of artificial intelligence and biomedical informatics. Description Logic, as essential layer of the Semantic Web stack, provides a logical formalism for ontology design, useful for concept representation and reasoning on the of domain-centred terminological knowledge.

An axiom is the fundamental modelling concept of a DL. Each axiom is defined by a logical statement composed by a set of concepts, individuals and their relationships. A terminological axiom is defined as

$$C \doteq D \mid C \sqsubseteq D \tag{1}$$

where C and D are concepts. A finite set of terminological axioms is known as T-Box T and is defined using the following descriptions. Note that $I \models C$ stands for *"I models C"*, where I is an interpretation function and C is a concept:

$$I \models C \sqsubseteq D\,if\,and\,only\,if\,C^I \subseteq D^I \tag{2}$$

$$I \models a\,only\,if\,C^I \subseteq D^I I \models T\,if\,and\,only\,if\,I \models \Phi\,for\,every\,\Phi \in T \tag{3}$$

An assertional axiom, representing concepts positively stated, is composed by a set of statements representing basic knowledge about individuals classified within the T-Box hierarchy. As with T-Box, an A-Box A is composed by a set of assertional axioms, according to these definitions:

$$I \models a : C \, if \, and \, only \, if \, a^I \in C^I \tag{4}$$

$$I \models (a, b) : R \, if \, and \, only \, if \, \left(a^I, b^I\right) \in R^I \tag{5}$$

$$I \models A \, if \, and \, only \, if \, I \models \Phi \, for \, every \, \Phi \in A \tag{6}$$

where a and b are individuals and R is a particular role. Given these formal definitions, a knowledgebase K is an ordered pair of T-Box and A-Box, defined as follows:

$$K = (T, A) \tag{7}$$

$$I \models K \, if \, and \, only \, if \, I = T \, and \, I = A \tag{8}$$

3 State of the art

Before starting with the design of a new ontology for the described problem, we carried out a study and evaluation of several related approaches that could help in the design of OBEliSK. The knowledgebases found that are closer to the approach we try to develop are described in the following points:

- **DEXMART**. EU funded FP7 project DEXMART [9] was launched in 2008 with an extension of 48 months. This project's key objectives are, among others, *i)* the development of original approaches for interpretation, learning and modelling of human object manipulation actions, and *ii)* the design of novel techniques for task planning for conferring the robotic system with self-adapting capabilities. In this project, handling knowledge and manipulation activities are modelled by means of linking actions to certain object categories.
- **GRASP**. EU-FP7 project GRASP [10] has the objective of designing a cognitive system capable of grasping and handling objects in open environments where unexpected events may occur. Data is managed through a semantic-based approach based on Cutkosky's taxonomy [11], in which different kind of grasps that a human hand is able to perform are categorized.
- **HANDLE**. Like GRASP and DEXMART, HANDLE [12] is also an European FP7 project lasting for four years. Its aim is to understand and replicate human object grasping and skilled hand movements using an anthropomorphic artificial hand by means of object affordances characterization for learning and replicating human handling tasks. Its database was designed for storing objects' models divided in graspable regions with an approach direction to the object and a default hand configuration for each applicable grasp strategy.
- **RoboEarth**. Developed by scientists from Philips, Eindhoven University of Technology and other four European universities, RoboEarth [13] is a robotic-oriented database designed as a cloud computing service with the

aim of making robots capable of learning new abilities from other robots by easing their communication. The database stores CAD models, point clouds and images from each object in order to retrieve information about its presence and location. It also stores semantic information about definition of actions to be executed for grasping each object, allowing to reuse this information on several hardware platforms.

As summarized in Table 1, our approach tries to improve some of the shortages found in the previous state of the art study. After an in-depth evaluation of the advantages and lacks shown by each of the studied projects, we have concluded that GRASP is the project that has more common objectives with our approach.

4 Knowledgebase design

Within the scope of the CogLaboration project, there is a need for modelling the entities to be handled by the robotic system. Instead of using a traditional relational database system, the decision of modelling the object taxonomy using semantic web based technologies provides the project with the ability of modify and expand the model without making any modification on previous developments. This is why, in our approach, the taxonomy was developed using OWL and entity data is stored as triples using RDF.

4.1 Object perception

An object representation is composed of multiple views which are essential for the identification and classification of the object by the perception module. The data acquired by the Kinect sensor is processed to find the possible location of the object to be recognized and then classified using the set of object models' partial views. As first step, a whole 3-D model of the object is captured and processed using both automated and manual methods. Once the model acquisition has been performed, the following phase consists on executing the model training process, which has been made over a set of partial views extracted from the complete object model. Since descriptors used for the classification are dependent on the specific viewpoint, we decided to generate a rich set of views for each object from different perspectives. Taking into account that is impractical to store these views (161 per object; 30 MB each) in a serialized form in the knowledgebase, views' file paths are stored instead. This is because transmission and deserialization tasks are highly time-consuming and is unacceptable to be used in *i)* the perception subsystem, intended for real-time operation, and *ii)* the robotic cognitive control subsystem, conceived for executing a fluent interaction.

Using this model description we are able to identify the set of properties that have to be considered for the object contextualization by the perception subsystem. Fig.1 shows the object perception-related features to be managed by the knowledgebase. Also, a brief description of each field is provided in Table 1.

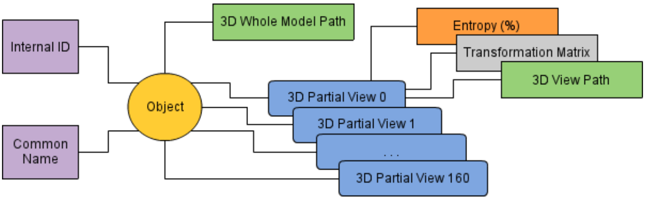


Fig. 1. Perception-related object properties. *Object* is the core element, having an *internal identifier* for its indexing withing the knowledgebase and a *common name* for the object's human-readable description. *Full 3-D model* file system path is also linked. For each *partial 3-D view* comprising the object model, its *entropy* (percentage of model shown in a particular view), *transformation* matrix (transformation between each view and the reference model) and *3-D view path* is also stored.

4.2 Object handling

Exchange properties. Besides the object visual properties representation, it is also crucial to store and manage the set of features describing the way in which each object can be handled during the handover phase of the exchange process. We have modelled the whole set of essential attributes within the knowledgebase, according to the requirements provided by the cognitive control layer regarding object grasp and delivery characteristic properties, in order to afford valuable information to this layer for achieving a fluent exchange process.

Each object is associated to a set of grasp postures and delivery strategies, defining different ways the robot can handle it. The set of identified features of these entities are described in the following sections. Moreover, in order to ensure a proper manipulation process for certain objects, we have introduced the concepts of *Motion Constraint* and *Symmetry Axis*.

Motion Constraint stands for the restrictions to be applied to some objects during their handling. For example, when handling a coffee mug, it is necessary to keep it on a vertical plane in order to avoid spilling its content. For this reason, when some motion constraints for the object to be handled are identified, we are able to establish its preferred orientation while is held and the tolerance range in which object's rotation should be kept. Furthermore, *Symmetry Axis* represents the axial symmetry of the object if there are any. This simple concept provides some useful information at the moment of grasping the object, so the control layer will know that the object appearance will remain the same at any position when rotated around its symmetry axis, easing the hand-to-object approaching phase.

Object affordances. We have also explored the concept of object affordances [14] within the task of modelling the ontology [15]. As stated in [16], classical approaches are based on visual properties of the instances for categorizing them. The taken approach is related to the concept of affordances and based on the idea of categorizing objects based on how are they used. According to Gibson's

Theory of Affordances [17], affordances can be seen as the sum of the properties of a situation, including agents, environment and objects, especially those that describe how they can be used to do something. For this reason, we have classified the object's classes according to their grasp area, taking into account the common way in which humans handle them, looking for a more comprehensive taxonomy that allows us to establish a clear model of relationship between objects and their actions concerning the exchange process.

Grasping phase. The set of grasps to be considered relies on the automatic grasping capabilities provided by the IH2 Azzurra robotic hand. These grasps are based on the taxonomy developed by Cutkosky and modelled under the class *GraspType*. As far as the work developed by Cutkosky is the grasp taxonomy, the modelling process using OWL is made straight from that one to our model, thanks to the hierarchical shape and the classification-oriented vocabulary respectively. At first, the whole grasping modes defined by Cutkosky were considered to be included in the design of our knowledgebase, but, in line with the robotic hand design and continuous improvements made during the project, we agreed on reducing the whole set of grasping modes to a set of primitives covering a wide range of them.

Each grasp is represented by a named individual and classified under a global class encompassing all of them. Although each individual grasp represents an object-specific configuration, the self-adaptability capabilities of the IH2 Azzurra hand allow to execute different grasp configurations for objects with similar shape. For example, two different objects having cylindrical shape and similar physical properties could be grasped using the same grasp mode. With the aim of properly model the concept of grasping action for the project's particular environment, we have extracted the initial set of concepts, as shown in Fig. 2, that help us to design and develop the grasping action modelling.

- *Object involved in the action.* We have to determine which kind of object is going to be handled during the exchange process. This property is the main element of each grasp model, in which the whole set of related properties will be linked.
- *Suitable grasp posture.* It is inferred according to the previously introduced taxonomy and adjusted to the set of grasping primitives capable to be executed by the robotic hand.
- *Grasp force level.* When grasping the object, the robotic hand controller has to recognize the instant when the object has been enclosed as expected. For this purpose, the level of force to be reached by the fingers is defined for each grasp posture.
- *Object approaching.* Before executing the grasping action itself, the robotic hand has to reach the object location. In this pre-grasping phase, the hand has to get close to the object in a suitable way. For instance, objects hanged on a wall will not be reachable from behind and this context would discard some grasp options. To tackle this situation, the model assigns an approach

direction to each grasp strategy in order to establish the proper way of approaching the object.
- *Robotic hand wrist pose.* Taking into account each object's shape, size, position and orientation, we have defined a desirable wrist pose for the robot when grasping each object. Using this information, the cognitive controller will be able to orient the wrist joint in the right posture in order to execute the grasp at the right region over the object's surface.

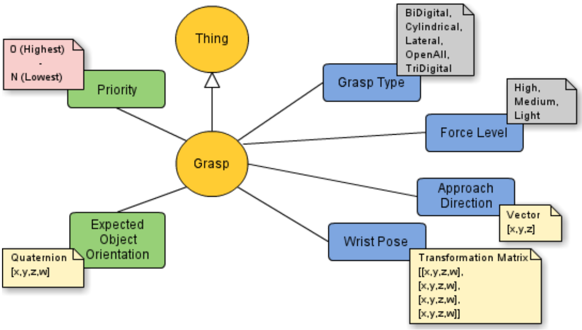


Fig. 2. Grasp model conceptualization. For this entity, *Grasp* is the core element. Each grasp strategy has a *grasp type*, a *force level*, an *approach direction* and a *wrist pose*.

During the integration phase of the knowledgebase subsystem with the cognitive control layer of the developed system as a whole, we have identified the necessity of distinguishing between different grasp strategies for the same object in order to fit the best suitable one for each situation. For this reason, we have introduced two concepts facing the adaptive capabilities of the knowledgebase, using reasoning concepts based on the feedback provided by previous exchange processes and real-time perception properties:

- *Grasp strategy priority.* By feeding the knowledgebase with the output generated during the execution of grasp actions for each object, we can use this feedback for prioritizing the best strategies to be used when grasping the object. For each object, a proper set of grasping strategies is sorted using this learned priority.
- *Expected object orientation.* We have represented the object's expected orientation when executing each grasp strategy. Using the object's position gathered by the perception system as input, the knowledgebase is able to provide the cognitive controller with grasp strategies whose expected orientation fits better with the perceived one, discarding those that are considered unsuitable according to the current positioning of the object to be grasped. Fig. 3 shows this concept in a more visual way.

In order to populate the project's knowledgebase with real data regarding different grasping strategies for the involved objects, researchers from SSSA designed

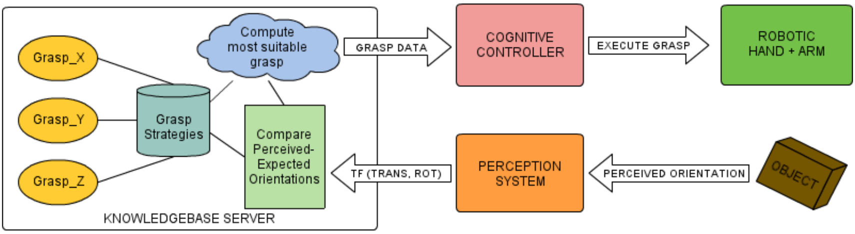


Fig. 3. Grasp strategy selection based on *perceived object orientation*. The *perceived object orientation*, provided by the Kinect sensor through the *perception system*, is sent by means of a *translation-rotation message*. Each *grasp strategy* expected orientation is compared among the perceived one. Our algorithm computes these inputs and provides the most suitable *grasping method*, that will be sent to the *robotic cognitive controller* which finally commands the physical movement execution on the *robotic setup*, composed of the KUKA LWR arm and the IH2 Azzurra hand.

and executed an experiment [18] based on performing a task of pick-and-lift each object during 30 times, using the robotic hand mounted on an able-bodied splint fixed to the experiment operator's arm. Orientation, distance of the hand from the target object and grasp type performed were varied and recorded at the instant of grasp using the *Vicon Motion Capture System*. Each trial was considered successful if the object was lifted at the correct distance without any movement with respect to the hand.

Delivery phase. Also, we considered the idea of improving the knowledgebase value for the project by extending the initial conception of a grasping database to a fully-featured handling knowledgebase. In this way, the inclusion of the handover-delivery concept was identified as essential, covering the second half of the object exchange process. The control system has to be provided with relevant data about the object handover, being capable to deliver the previously grasped object to the recipients in a fluent and natural way. Keeping this goal in mind and taking advantage of the analysis and development phase for the previous grasp conceptualization, the following key concepts, also shown in Fig. 4, were identified for describing the delivery action:

- *Object to be delivered.* We have to determine which kind of object is going to be delivered to the receiving partner. This property will be the main entity of each delivery model in which the whole set of related properties will be linked.
- *Grasp strategy used.* In order to take advantage of the system's reasoning capabilities and fit the behaviour depending on the current situation, each delivery strategy has to be linked with the grasp previously used when grasping the object. Using this property, we can determine which delivery methods are available.
- *Expected human hand posture.* In order to fine-tune the way of presenting the exchanged object to the human, based on the work performed in [19],

the refined reasoning mechanism implemented for selecting the most suitable delivery method was designed looking for a way of orientating and positioning the object when held by the robotic hand, taking into account human's hand orientation and the grasping strategy previously used. The decision of using a fixed set of orientations for symbolizing the receiver's hand posture is given by the unfeasibility of observing its concrete grasping posture, since the fingers get positioned too late during the exchange action, making the operation non-compliant with the fluency objectives. Also, the *stop* posture has been also added to the gesture list for providing the option of cancelling an exchange during its execution.

- *Approach direction.* Along the same line as the grasp conceptualization is described, the approaching phase involves, in this case, the approximation of the object (held by the robotic hand) to the human partner's hand. Each delivery strategy has a default approach direction where the object has to be placed.
- *Robotic hand wrist pose.* Again, the wrist pose of the robotic hand is an important parameter to be used when establishing a proper position and orientation for delivering the exchanged object. This attribute is combined with the *Motion Constraint* property, introduced above. Due to particular object's properties, the wrist pose adopted during the handover could be conditioned, so the knowledgebase has to take these restrictions into account and apply inference techniques about them for providing a suitable delivery strategy.
- *Strategy selection preference.* The priority attribute is also useful for sorting each strategy according to its results when applied on previous delivery procedures, in the same way as presented for the grasp strategy conceptualization.

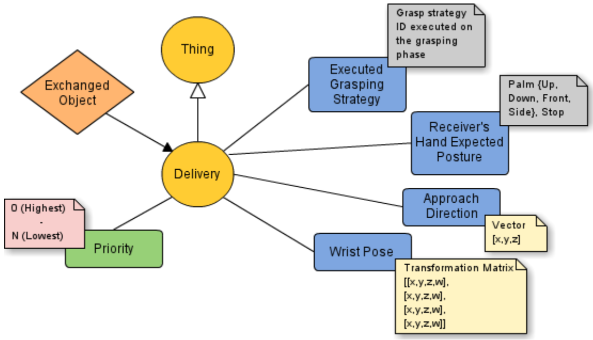


Fig. 4. Delivery model conceptualization. *Delivery* is the core element here, and each delivery strategy is linked with a previously *executed grasping method*. Also, the *receiver's hand expected posture* is decisive when selecting the most suitable strategy in order to provide the robotic hand with a proper *approach direction* and *wrist pose* for passing the object to the receiver.

5 Results

In this paper we have presented a novel approach for representing the object exchange process between robotic and human agents in a fluent way. Our contribution is compared with the state of the art approaches in Section 5.1. The developed knowledgebase management tool is also described in Section 5.2.

5.1 Comparative analysis

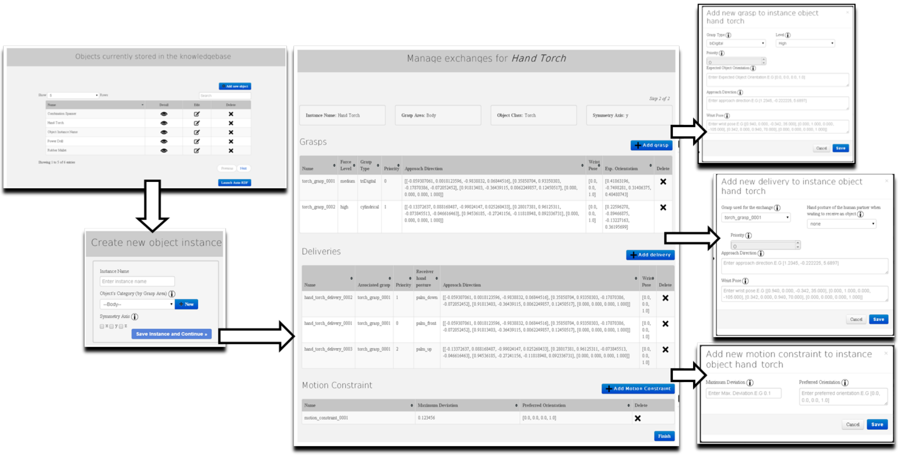
The main advantage of our knowledgebase design is that provides the required set of mechanisms that makes it suitable for working together with artificial vision and cognitive control modules, providing the robot with the skills needed for achieving a fluent object exchange process by means of concepts from computer vision, artificial intelligence and cognitive neuroscience. Besides the abovementioned grasping capabilities, the object handover process has also been incorporated, providing all the required properties for this action while taking into account the potential motion restrictions to be applied. Moreover, the developed knowledgebase counts on an automatic learning system that uses the feedback provided by the cognitive controller. This knowledge is continuously growing with each exchange process executed, allowing the system to be smarter by means of learning which grasping and delivery methods are more natural and fluent for each object type. Finally, another advantage of our knowledgebase among others is the standardization of the object exchange model, providing a ground ontology that may be re-used on new systems with common goals. The ad hoc developed data management tool, provided as a web-application, allows and eases the task of incorporating data to the knowledgebase in a straightforward way, without requiring any prior knowledge neither on semantic technologies nor its internal design.

Table 1. Comparative analysis of different capabilities of our approach versus other representative projects introduced in Section 3.

Functionality	OBEliSK	DEXMART	GRASP	HANDLE	RoboEarth
Integrated perception	✓	✗	✓	✗	✓
Human-like handling	✓	✓	✗	✓	✗
Obstacle avoidance	✓	✗	✓	✗	✓
Obstacle avoidance	✓	✗	✓	✗	✓
Multi-mode grasping	✓	✓	✓	✓	✓
Object handover	✓	✗	✗	✗	✗
Standarized data store	✓	✗	✓	✗	✓
Motion constraints mgt.	✓	✓	✗	✗	✗
Learning capabilities	✓	✓	✗	✗	✗
Data management tool	✓	✗	✗	✗	✗

5.2 Interface for knowledgebase data management

Data processing and storage in this sort of databases is not trivial. Having a large amount of information and a defined ontology, it is mandatory to fully respect the relational integrity restrictions between entities and their properties, avoiding, for example, inconsistency problems on the data schema or potential issues when executing recursive queries. Semantic-ontological data management is usually done through semantic-oriented tools, such as Protégé [20], that allows data model creation and instance population. Although this is not a bad option, low-level knowledgebase data management may be a difficult issue for anyone without previous experience in this field. With the aim of ease this task, a utility has been developed focused on offering the simplest way to populate the knowledgebase with the results of the studies on object handling carried out by project's partners. Moreover, it is intended to publish this utility on the CogLaboration's website in order to make it available for other researchers to take advantage of our results. This utility, as shown in Fig.5, consists of a web-



Show 5 Rows

Name	Detail	Edit	Delete
Combination Spanner	👁	✎	✕
Hand Torch	👁	✎	✕
Object Instance Name	👁	✎	✕
Power Drill	👁	✎	✕
Rubber Mallet	👁	✎	✕

Showing 1 to 5 of 6 entries

Object's Category (by Grasp Area): --Body--

Symmetry Axis: ☐ x ☐ y ☐ z

Instance Name: Hand Torch
Grasp Area: Body
Object Class: Torch
Symmetry Axis: y

Grasps

Name	Force Level	Grasp Type	Priority	Approach Direction	Wrist Pose	Exp. Orientation	Delete
torch_grasp_0001	medium	triDigital	0	[[-0.059307061, 0.0018123596, -0.9838832, 0.06844516], [0.35850704, 0.93350303, -0.17870386, -0.072052452], [0.91813403, -0.36439115, 0.0062249857, 0.12450517], [0.000, 0.000, 0.000, 1.000]]	[0.0, 0.0, 1.0]	[0.41863196, -0.7498281, 0.31406375, 0.40480743]	✕
torch_grasp_0002	high	cylindrical	1	[[-0.13372637, 0.088168487, -0.99024147, 0.025260433], [0.28017381, 0.96125311, -0.073845513, -0.046616463], [0.94536185, -0.27241156, -0.11818948, 0.092336731], [0.000, 0.000, 0.000, 1.000]]	[0.0, 0.0, 1.0]	[0.22596278, -0.89466875, -0.13227163, 0.36195699]	✕

Deliveries

Name	Associated grasp	Priority	Receiver hand posture	Approach Direction	Wrist Pose	Delete
hand_torch_delivery_0002	torch_grasp_0001	1	palm_down	[[-0.059307061, 0.0018123596, -0.9838832, 0.06844516], [0.35850704, 0.93350303, -0.17870386, -0.072052452], [0.91813403, -0.36439115, 0.0062249857, 0.12450517], [0.000, 0.000, 0.000, 1.000]]	[0.0, 0.0, 1.0]	✕
hand_torch_delivery_0001	torch_grasp_0001	0	palm_front	[[-0.059307061, 0.0018123596, -0.9838832, 0.06844516], [0.35850704, 0.93350303, -0.17870386, -0.072052452], [0.91813403, -0.36439115, 0.0062249857, 0.12450517], [0.000, 0.000, 0.000, 1.000]]	[0.0, 0.0, 1.0]	✕
hand_torch_delivery_0003	torch_grasp_0001	2	palm_up	[[-0.13372637, 0.088168487, -0.99024147, 0.025260433], [0.28017381, 0.96125311, -0.073845513, -0.046616463], [0.94536185, -0.27241156, -0.11818948, 0.092336731], [0.000, 0.000, 0.000, 1.000]]	[0.0, 0.0, 1.0]	✕

Motion Constraint

Name	Maximum Deviation	Preferred Orientation	Delete
motion_constraint_0001	0.123456	[0.0, 0.0, 0.0, 1.0]	✕

Grasp Type: biDigital
Level: High
Priority: 0
Expected Object Orientation
Approach Direction
Wrist Pose

Grasp used for the exchange: torch_grasp_0001
Hand posture of the human partner when waiting to receive an object: none
Priority: 0
Approach Direction
Wrist Pose

Maximum Deviation
Preferred Orientation

Fig. 5. Web-based knowledge management tool. *Left side of the image -Top*: Current object list stored in the knowledgebase. *Bottom*: Creating a new object instance. *Middle side of the image*: Exchange properties management view. *Right side of the image - Top*: Adding a grasping strategy and a delivery one to an existing object. *Bottom*: Defining a motion constraint to be applied when handling an object.

application acting as interface between the user and the triplestore, where the ontology data is saved. In this way, the person in charge of entering data on the knowledgebase will not have to deal with complex terms of the ontology, such as relationships, hierarchies or entities membership. All data is provided through the user-oriented interface and will be automatically processed and arranged in order to include it into the system. For this input processing, the module called

Auto-RDF has been developed, composed by a set of Python-based scripts acting as middleware layer, retrieving input data from the web interface and the vision-perception data files (models, views, etc.) for adapting it to RDF triples format. Then, these triples are linked following the existing structure of classes and properties for, finally, adding them to the ontology in the OpenLink Virtuoso triplestore.

6 Conclusions

This paper has introduced the work performed for providing the CogLaboration's robotic system with the knowledge on objects' properties and the particularities of their exchange process. We have taken an innovative approach by means of semantic-ontological technologies which provides the project with interesting and innovative results. The developed knowledgebase meets the expectations and overcomes them, because we extended the initially proposed grasping model to a complete exchange one by means of the inclusion of object handover concepts.

Additionally, some other concepts were included when identified as useful for improving the object handling fluency of the robotic system, including its adaptability to each particular situation in non-deterministic scenarios. Regarding the communication and feeding of data from the handling knowledgebase to the robotic controller, the developed techniques, provided in a structured and complete interface of query services, allows the system to adapt the strategy selection to the specific situation, while taking advantage from the continuous learning process that improves the algorithm in charge of selecting the most advisable strategy for grasping, handling and delivering each object. We are concerned with the interoperability needs of this task among the rest of project developments, so we have dedicated a considerable amount of our efforts to provide a simple, understandable and comprehensive interfaces for both inputs and outcomes regarding the knowledgebase. Specifically, we have worked in parallel with the developments done within the cognitive controller in order to familiarize with its architecture and future needs. Also, the input provided from project's partners from their studies on exchange process, motion capture trials and experimental reports were highly valuable contributions for the knowledgebase design, development, population and subsequent refinement. The introduced system will be used within the project's final human-robot interaction experiments.

7 Acknowledgments

This work was supported by the CogLaboration project within the 7th Framework Programme of the European Union, Cognitive Systems and Robotics - Contract Number FP7-287888. See also http://www.coglaboration.eu/ .

The authors would also like to thank Dr. Marco Controzzi and Dr. Rubén Casado for their valuable comments.

References

1. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? IEEE Intelligent systems **14**(1) (1999) 20–26
2. Presutti, V., Gangemi, A.: Content ontology design patterns as practical building blocks for web ontologies. In: Conceptual Modeling-ER 2008. Springer (2008) 128–141
3. Codd, E.F.: A relational model of data for large shared data banks. Communications of the ACM **13**(6) (1970) 377–387
4. Angles, R., Gutierrez, C.: Survey of graph database models. ACM Computing Surveys (CSUR) **40**(1) (2008) 1
5. Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., Wilkins, D.: A comparison of a graph database and a relational database: a data provenance perspective. In: Proceedings of the 48th annual Southeast regional conference, ACM (2010) 42
6. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific american **284**(5) (2001) 28–37
7. Brickley, D., Guha, R.V.: Resource Description Framework (RDF) Schema Specification 1.0: W3C Candidate Recommendation 27 March 2000. (2000)
8. McGuinness, D.L., Van Harmelen, F., et al.: OWL web ontology language overview. W3C recommendation **10**(10) (2004) 2004
9. Siciliano, B.: DEXMART. DEXterous and autonomous dual-arm/hand robotic manipulation with sMART sensory-motor skills: A bridge from natural to artificial cognition. http://www.dexmart.eu (2008–2012)
10. Kragic, D.: GRASP. Emergence of Cognitive Grasping through Introspection, Emulation and Surprise. http://www.csc.kth.se/grasp (2008–2012)
11. Cutkosky, M.R.: On grasp choice, grasp models, and the design of hands for manufacturing tasks. Robotics and Automation, IEEE Transactions on **5**(3) (1989) 269–279
12. Perdereau, V.: HANDLE: Developmental pathway towards autonomy and dexterity in robot in-hand manipulation. http://www.handle-project.eu (2009–2013)
13. Tenorth, M., Perzylo, A.C., Lafrenz, R., Beetz, M.: Representation and exchange of knowledge about actions, objects, and environments in the roboearth framework. Automation Science and Engineering, IEEE Transactions on **10**(3) (2013) 643–651
14. Varadarajan, K.M., Vincze, M.: Knowledge representation and inference for grasp affordances. In: Computer Vision Systems. Springer (2011) 173–182
15. Varadarajan, K., Vincze, M.: Ontological knowledge management framework for grasping and manipulation. In: IROS Workshop on Knowledge Representation for Autonomous Robots. (2011)
16. Stark, M., Lies, P., Zillich, M., Wyatt, J., Schiele, B.: Functional object class detection based on learned affordance cues. In: Computer Vision Systems. Springer (2008) 435–444
17. Gibson, J.J.: The theory of affordances. Hilldale, USA (1977)
18. Controzzi, M., Montagnani, F.: Experiments on hand postures for object grasping. Internal report, The CogLaboration Project (FP7/2007-2013 grant agreement number 287888) (2014)
19. Rasines, I., Remazeilles, A.: Human hand pose recognition for non-verbal communication. Public deliverable, The CogLaboration Project (FP7/2007-2013 grant agreement number 287888) (2014)
20. Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A.: The Protégé OWL plugin: An open development environment for semantic web applications. In: The Semantic Web–ISWC 2004. Springer (2004) 229–243