Using Semantic Technologies to Describe Robotic Embodiments

Alex Juarez
Dept. of Industrial Design
TU Eindhoven
Den Dolech 2
Eindhoven, Netherlands
acordova@tue.nl

Christoph Bartneck
HIT Lab NZ
University of Canterbury
Private Bag 4800
Christchurch, 8140
christoph.bartneck@
canterbury.ac.nz

Loe Feijs
Dept. of Industrial Design
TU Eindhoven
Den Dolech 2
Eindhoven, Netherlands
l.m.f.g.feijs@tue.nl

ABSTRACT

This paper presents our approach to using semantic technologies to describe robot embodiments. We introduce a prototype implementation of RoboDB, a robot database based on semantic web technologies with the functionality necessary to store meaningful information about the robot's body structure. We present a heuristic evaluation of the user interface to the system, and discuss the possibilities of using the semantic information gathered in the database for applications like building a robot ontology, and the development of robot middleware systems.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.5 [**Information Interfaces and Presentation (I.7)**]: User Interfaces

General Terms

Design, Measurement

Keywords

Collaborative Systems, Robot Database, Robotic Embodiments, Semantic Technologies

1. INTRODUCTION

Semantic Technologies (ST) is a term used to denote the family of techniques and tools developed to provide *meaning* to the vast body of information that is already available in digital format. These tools model knowledge and *link* together several heterogeneous *resources*, such that the *information* provided by these resources can be processed automatically by *agents*. Common applications are the construction of knowledge bases and expert systems [8, 12].

ST have also found application in the field of robotics. Examples of the feasibility and applicability of these technologies in the field are the knowledge representation used for different robotic devices, and the description of experience gained through robotic learning episodes.

In this paper we present our approach to the use of semantic technologies to build a database of robot descriptions. Focusing on the robot's body structure has the advantage that the characteristics captured in the descriptions are easily and more objectively verifiable, compared to descriptions based in behaviors or software capabilities (e.g. collaboration, sociability, etc.). The information gathered has great value for the robotics community, as it can then be used by different kinds of users for multiple purposes. For example, the robot programmer looking at hardware properties to create new devices, might use this information to find out the trend of robotics development. The robot hobbyist interested in acquiring commercial robots to clean up a house, could obtain in a quicker and easier way information to assist in making a better decision.

In the following sections we explore past applications of ST to robotics, and the need to describe the physical structure of robots (Section 2), as well as the relevance this has for the HRI community (Section 2.1). Thereafter, we present our approach to solving the problem of specifying the robot structural properties with the use of ST (Section 3), and a prototype implementation called RoboDB (Section 4). Finally we discuss the possibilities of using the information gathered to build a robot ontology, and in the development of robot middleware.

2. DESCRIBING ROBOTS AND THEIR CAPABILITIES

A survey through the literature reveals a body of robotics research where ST have found application. Examples of this are the works of Chella et al. [5] where an ontology was used to describe the environment where a mobile robot moves. Using this information, the robot could in principle make decisions about the proper way to navigate through space. Mendoza et al. [14] used ST to represent and manage the relationship between entities that were recognized by robotic vision software. Yanco and Drury [24] created a taxonomy for human-robot interaction (HRI) to allow the comparison of different HRI systems. An updated taxonomy presented in [25] included high level concepts like interaction roles and *robot morphology* as categories used to classify robots.

The majority of approaches that use ST have created knowledge representations *for* robots, while only a few have produced knowledge *about* robots. Little work has focused in using ST to describe the *robot embodiment*, concentrating instead on the description of the environment where the robot 'lives'. In this paper we use the term *robot embodiment* referring to the *physical embodiment* of a robotic device. Physical embodiment is a notion stating that "embodied systems need to have a physical instantiation" considering the robot components as part of a "physical grounding" needed to connect the robot with the environment [26]. A full analysis of the different definitions of robot embodiment in literature is out of the scope of this paper, however in Section 5 we briefly discuss the implications of the definition used in this paper to our current system implementation.

The need for a proper embodiment description is also an ongoing topic of discussion in the robotics community, specially among researchers focusing on the development of *robotic middleware*. The definition of middleware describes it as a "class of software technologies designed to help manage the complexity and heterogeneity inherent in distributed systems" (Bakken, D. [2]). Since robots are intrinsically distributed systems, robotic middlewares must find a way to appropriately describe an embodiment that is, by nature, heterogeneous. Once this is achieved, control software, remote robotic control software, and robotic tele-presence interfaces becomes much more reusable and manageable.

It is worth noticing that, even though each robotic middleware uses its own description method to determine the capabilities of a robot, there is still no standardized embodiment representation. Standardization is not only important for robot programmers to reuse their code, but also for content creators to more efficiently replicate their work to different embodiments. Therefore, any effort that brings the community closer to a standard description will be very worthwhile.

2.1 Relevance to the HRI community

Drury and Yanco [24] recognized the importance of the robot embodiment design characteristics (e.g. available sensors vs. provided sensors, sensor fusion, autonomy, etc.) to define individual HRI systems, and allow for their comparison. We believe that a structured body of information about robotic systems would allow not only to compare different social robots, but also to *monitor the progress of the community* by a) establishing trends on the development of social robots, b) establishing benchmarks for systems comparison, and c) grouping robots by quantifiable, material characteristics.

The need for access to information about robot embodiments is further stressed by Quick et al [19]. They state that the physical characteristics of robotic devices are essential to interpret the results of any study in the real world. However, many of the physical features of a robot are not visible with the naked eye (as is the case with wireless communication or skin sensing), which might result in inaccurate or misleading conclusions. This underlines the importance of a repository where also the non-obvious physical properties are stored.

But the need for such source of information is not restricted to scientific use. Burghart and Haeussling [4] argue that the common robot user can also benefit from making available the embodiment characteristics in a easy way.

They maintain that one of the requisites for effective human-robot collaboration, is that the human partner possess specific knowledge in how to handle the robot, its hardware constraints and the restricted set of senses available and applicable for a specific task.

To illustrate an application scenario of a embodiment properties repository, consider the case of a person that wants to buy a robot to help him clean his house. He would like to be able to control the robot from his computer, but also to give the robot voice commands. He would like the robot to be *safe* but at the same time, to help him clean the house as quickly as possible.

By using an embodiment description database with thousands of robots, the user could ask for those robots *with microphones* so that they can hear his commands. He could also narrow down his options by asking which of those robots also have either a *vacuum cleaner* or a *mop* to clean the floor. He could also decide to look for robots with *hands* so that it can also clean other surfaces. Furthermore, the user can also request the maximum and average speed of the robots and determine if they will be safe to run around home. He can also figure out if these robots have *wireless networking* as that would allow them to be hooked up to his tablet or PC for remote control. Armed with a reduced set of robots that fit his basic requirements, he can make an *informed decision* on which robot is best for his home.

We argue that the structured information gathered using ST can have an impact on the way collaborative, interactive systems are designed, built, and used. As in our example, the common robot user would easily find the information he needs about specific robots (or sets of robots). For robotics experts, it will keep them up to date in the latest developments in the field, but more importantly, it will allow the reuse of interaction patterns, as robotics content creators can create scripts for robot actions that can run in multiple robots with the same (or similar) embodiment description.

3. METHODOLOGY

ST provide mechanisms to represent available knowledge about robotic embodiments in a reusable, machine-friendly way. In this section we briefly review the theoretical foundation for ST (Section 3.1), followed by an outline of the system requirements and design decisions taken in our approach (Section 3.2).

3.1 Background knowledge

Description logics (DL) are a family of formal knowledge representation languages. DL are used in several disciplines where there is a need to code concepts of a domain, and perform formal reasoning about them. DL's are of particular importance in providing the logical formalism for Ontologies and the Semantic Web [21].

The basic terminology of DL consists of *concepts*, *roles*, and *objects*. Objects denote entities of our world with characteristics and attributes. Concepts are interpreted as sets of objects. Roles are interpreted as binary relations on objects or concepts. These definitions have a direct translation into ontology terminology, where concept is a synonym of *class*, role is a synonym of *property*, and object is a synonym of *individual*.

In DL, the term TBox (*terminological box*) refers to sentences describing relations between concepts. The term ABox (*assertional box*) refers to sentences describing relations be-

tween objects and concepts, i.e. the set of grounded sentences. In ontology terminology, the TBox corresponds to the set of classes and properties while the ABox corresponds to the set of facts defined for an ontology. The union of ABox and TBox is generally known as *knowledge base.*

The body of knowledge in a knowledge base is expressed by sets of triples, formed by a role connecting two objects or concepts. The formal representation used by modern ST to model these relations and encode knowledge is called *Resource Description Framework* (RDF). Several derivations of this representation exist with varied levels of expressiveness. The most widely spread are *RDF-Schema* (RDFS) and *Web Ontology Language* (OWL).

Semantic annotations (SA) is a mechanism to encode knowledge in ST. Two of the key benefits of using semantic annotations is the enhancement of information retrieval and the improved interoperability. Information retrieval is improved by enabling structured searches and exploiting the inference capabilities of ST. Interoperability is improved by providing a common -language- framework to integrate information from heterogeneous sources [22].

3.2 Describing the robot physical structure

When designing a system to capture the characteristics of robot physical embodiments we must establish a set of basic requirements to meet. In our approach, the basic set of requirements is:

1. An attractive feature of robotic hardware is the facility to modify such hardware by adding or eliminating components. Therefore, the system must be flexible enough to cope with the high reconfigurability of robotic systems.

2. The system must be usable by both scientists and robot users. Usability will be measured with respect to the easiness with which the user can store and retrieve information from the system.

3. The data storage must be accessible by human users and machines alike, and the access mechanisms must provide a syntax that is easy to learn and remember.

4. It is hardly possible that a single person or even a small group of people can gather information about all available robotic embodiments. Therefore, the system must enforce collaboration and discussion between the robotics community and general public. The community itself should be in charge of contributing and maintaining the information in the system.

The last requirement has a strong impact on the design of our system. One alternative to achieve international collaboration from the robotics developers and users is through the use of the Internet. The *Web* has shown repeatedly its power to make information available with the smallest latency, a feature extremely important given the current rate of development of robots. It would also allow virtually everyone to have a say with respect to definitions and concepts produced by the system, encouraging communication, discussion, and collaboration.

With these requirements in mind, it seems natural to adopt a *web-based* solution to describe robot embodiments. ST have been traditionally linked to information technologies and web development, originating what is known today

as *Semantic Web Technologies* (SWT). These technologies cover several layers of abstraction, from the low-level resource encoding (e.g. URI, RDF) to the inference, logic and proof (e.g. Ontologies- OWL) presented in the previous section. These standards have been adopted and implemented in several 'flavors', into what constitutes today a wide and robust framework for *smart web* software development.

The main operations in RoboDB are the storage of new information about robot embodiments, the retrieval of stored information, and the modification of existing ones.

Adding new information to the data-base is a guided process where the system user must first identify the robot embodiment with a unique name. This is followed by requests for information about the robot sensor, actuators and other parts, as well as the connections between them. This is the crucial step that will create the embodiment description and capture the structural properties of the robot.

On each step of the process, semantic annotations are automatically generated and stored in the database. We are aware that semantic annotations might be hard to read by non-expert users. Furthermore, it might be hard to detect errors, or inconsistencies in the embodiment description by just looking at semantic codes. Thus, the annotations created by the system are used to generate an XML-based graphical description, that provides robot users with a picture of the physical structure of the robot. This graphical description is presented to the system user on each step of the process (See Figure 3). Both the annotations and the XML-based graph are stored in the database once the process is completed.

Retrieval of robot embodiment descriptions can be achieved in two ways. The human user can query the database using a web interface. Both the semantic annotations as well as the XML code used to create the graphical descriptions can be retrieved. A low level API is also available through the web, for access to other automated systems. The XML format is compliant with a recommendation submitted for consideration to the MPEG-V standardization initiative [7, 16].

Modification of embodiment descriptions follows the same guided approach used to store new information. Although there are different approaches to modify the embodiment description,the guided process is an intentional design decision to present the system user with a familiar way to change the stored data.

4. RESULTS

In this section we present the results in two parts. The first one explains the implementation of the prototype for RoboDB, a database that stores meaningful, structured information about robotic embodiments. The second part presents the heuristic evaluation of the prototype.

4.1 RoboDB

We developed a prototype system called RoboDB to store structured information about robot embodiments (Figure 2 shows the GUI of the prototype). After considering several implementation alternatives, we narrowed down our options to a) Protegé, an open-source ontology editor and knowledge acquisition framework, b) Semantic Mediawiki, a semantic web collaboration tool, and c) developing our own home-brewed SWT solution.
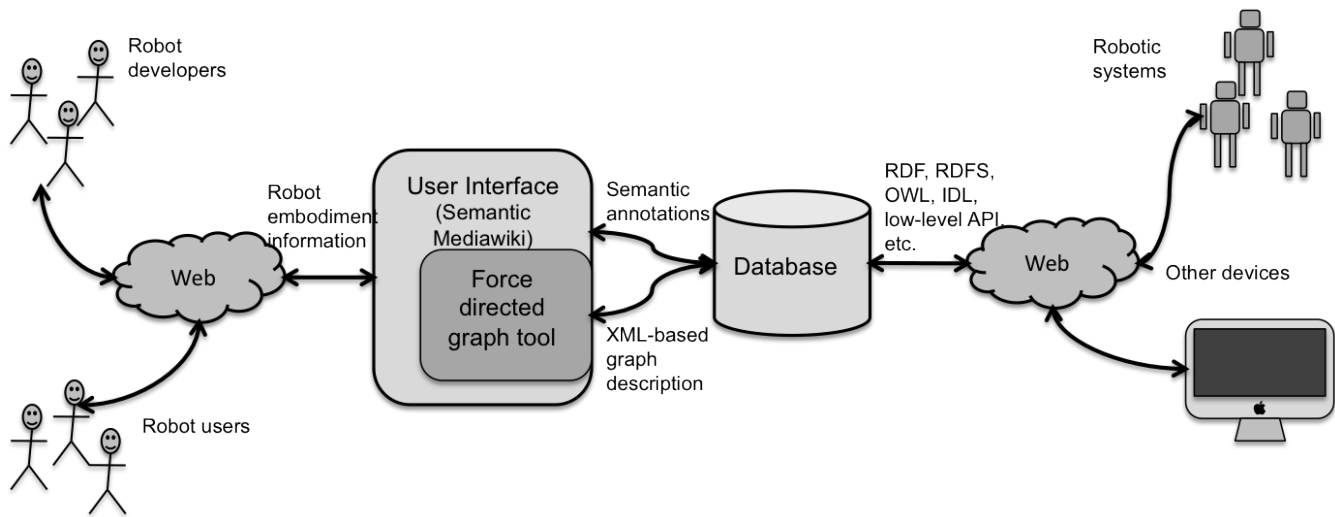
Figure 1: Overview of the system prototype

Figure 2: Graphical user interface of the RoboDB website.

Protégé [8] is an application used to model and build knowledge bases. It is considered a state of the art tool in semantic modeling and provides extensive set of features. However, Protégé requires a considerable learning effort to use it efficiently. Furthermore, despite providing a series of web plug-ins, Protégé still has to make its way on the web, as few web applications make use of it.

Semantic Mediawiki [11] is an open-source extension to the widely popular Mediawiki content management system used by numerous collaborative web applications like Wikipedia. The advantage it offers are the familiarity many users around the world have with the system, its inherent collaborative, community-oriented nature, the annotation-oriented approach to store information. The disadvantage of this system is its limited set of features compared to Protégé, and the lack of a proper semantic reasoning engine.

Building our own SWT solution would have the advantage of more control over the development of the features needed. However, it would take a considerable amount of effort before such solution would be at the same level of completeness and usability of either Protégé or Semantic Mediawiki.

Following the requirements outlined in Section 3.2, we decided to implement a hybrid solution using the Semantic Mediawiki system as a front-end for RoboDB, while at the same time, adding our own mechanism to create the graphical description, and implementing a method to export the embodiment descriptions into a format that can be further processed by external tools like Protégé. Figure 1 shows the outline of the system.

Semantic mediawiki provides a mechanism to produce semantic annotations on textual information. The syntax of these annotations is based on the popular Mediawiki syntax. Semantic annotations are transformed to an internal representation used to generate the graphical description of the robot embodiment. The user interface is further enhanced with browsing and annotation capabilities provided by the Halo extension [17] Finally, the annotations are stored in the database.

The graphical description is drawn using a force-directed graph method [6], and its implementation is loosely based in the approach by McCullough [13] using the Processing Java library. Further additions to the prototype system are the ability to export single or multiple robot descriptions to RDF and XML format, as well as the conversion between XML graphical description and semantic annotations. The front-end system has been enhanced with simple statistics and example queries on embodiment properties, for example, the robot distribution per country, size of TBox and ABox, queries on the number of existing legged robots, etc.

4.2 Knowledgebase creation

One of the outcomes of RoboDB is the construction of a knowledge base with the information contributed by robot developers and users. In this section we present some statistical figures extracted from the prototype implementation. These figures are not representative of the current state of robotic embodiments available worldwide, but they should be taken as a proof of concept and example of the kind of information and queries that could be asked to the database.

At the moment of writing, the prototype contained information about a set of 30 robot embodiments. The TBox
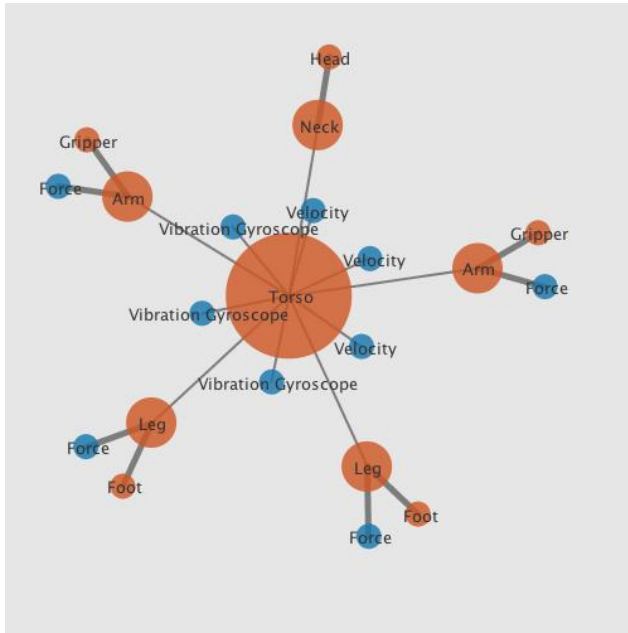
Figure 3: Example of a forced graph embodiment description. Blue nodes represent sensors. Orange nodes represent other robot components.

Boolean	Code	Date
Email	Length	Mass
Number	Page	Record
String	Temperature	Text
URL	Velocity	

Table 1: Data types currently defined in RoboDB

of the prototype had 54 classes and 70 properties. 14 data types are already defined (See Table 1). The distribution of properties by data types can be seen in Table 2.

The ABox of the prototype had 481 facts already stored, with each embodiment description containing between 10 and 26 facts. Examples of facts are the number of legs, arms, maximum speed, weight, etc. Table 2 shows the distribution of facts per data type.

Data type	Property count	Facts count
Number	52	376
Length	4	28
Mass	1	12
Velocity	1	3
URL	1	30
Date	1	32

Table 2: Properties and facts distribution by data type

As examples of the kind of queries that can be made to RoboDB, the top 5 most popular robot parts (comprises both sensors, actuators, and other robot parts) are shown in Table 3, while the most unique (the least frequent) are shown

in table 4. While these queries are based on facts about the robots, it is also possible to query *relationships* between robot components, e.g. torsos that have heads connected to them, or robots whose eyes include a camera for vision, etc.

Part name	Robot count
Torso	28
Head	27
Eyes	21
Camera sensor	18
Arms	18

Table 3: Most popular embodiment parts

Part name	Robot count
Tentacle component	1
LIDAR sensor	1
Trunk component	1
Barcode scanner	1
Active 3D IR camera sensor	1

Table 4: Least frequent embodiment parts

4.3 Heuristic evaluation of the prototype interface

We believe that part of the long term success of RoboDB will depend on the user experience at the moment of adding, modifying, and retrieving information about robot embodiments. Therefore, we conducted a heuristic evaluation of the current prototype with the collaboration of a small group of four HRI and systems design experts following the principles in [15]. Based on the comments from these experts we took the following design decisions:

a) Provide the user with lists of the most used components and sensors already available in the database. This creates shortcuts to construct the embodiment description based in what other users have already constructed. It must be noted that the user still has the ability to create new components 'on-the-fly' should he need it.

b) Subdivide the process of adding new information into consistent steps. The users are asked to input first the information about robot components and their connections. Only when this step is completed, the robot sensors are can be added. This reduces the memory load on the user, as he is not required to input all the information about the robot structure at once.

c) Use the same guided approach for the modification of existing embodiment descriptions. As a result, the system as a whole gains in consistency and the user can quickly accommodate to the process by repetition.

d) Include the graph preview of the robot on each step of the process of adding or modifying an embodiment description. This gives appropriate, updated feedback to the user, allowing him to spot errors at any stage of the process.

Figure 4 illustrates the applications of these principles to one of the steps in the process of adding a new robot embodiment: The user is clearly informed of the amount of steps to complete the addition of new information. The user is presented only with the available options to connect com-
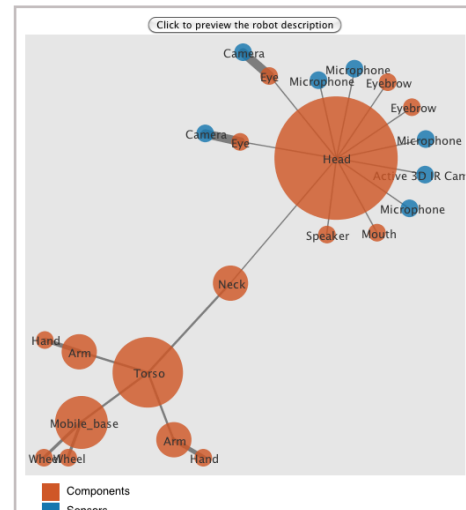
Figure 4: Illustration of some of the refined user interface after heuristic evaluation by experts

ponents between themselves, reducing the risk for errors in complex robot descriptions.

The connections that are already made, are shown in the graph display minimizing the memory load. At every moment the user can decide to delete or add connections between components and the result is reflected immediately, increasing the awareness of the state of the system.

5. DISCUSSION AND FUTURE WORK

In this section we discuss the limitations of our approach and present possible applications for the information gathered in RoboDB.

5.1 Implications of the embodiment definition

Embodiment is a term with several competing definitions in robotics. In its current implementation, RoboDB uses the notion of *physical embodiment* as working definition for robot structure descriptions. We have intentionally constrained further this notion to focus on the physical body parts of the robot. The advantage of this working definition is that it allows us to focus on embodiment characteristics that are (to some degree) verifiable and quantifiable.

However, we are fully aware that this is also a limitation of the system: robot embodiments cannot be described only in terms of its physical components, but also in terms of behaviors and abilities. For example, it cannot be assumed that just because a robot has legs, it also has the ability to walk, even though a "walking" capability is integral to a description of the robot "self". We intend to build upon our first results and address the challenge of extending the system to create a richer description in terms of physical characteristics, behaviors and capabilities.

5.2 Building the RoboDB community

We are aware that much of the success of RoboDB depends on a community effort to contribute information to it.

At the local level, RoboDB is collaborating with RoboNED, a robotics organization from the Netherlands that gathers robotics scientists from industry and academy. Through this initiative, partners from RoboNED will become users and contributors of RoboDB turning it into the information point for robotics in the Netherlands. At international level, we already initiated contacts with organizations like AIST and ATR in Japan, which already expressed their interest in joining RoboDB as contributors. This contacts will gradually increase through presentations and tutorials at key conferences and events.

5.3 Using robot embodiment descriptions to build a robot ontology

SWT have been developed as the *Web* alternative to Semantic Technologies. It is only natural that the information encoded in semantic annotations can be used to build a robot ontology.

Ontologies have found application in knowledge modeling in several fields. A well known success story is the application of ontology reasoning to genetics with the Gene Ontology [1]. Historically, advances in gene sequencing had been hindered by the different ways used by scientists to describe and conceptualize shared biological elements of organisms. Here is where ontology-based approaches have contributed to numerous key findings by creating a unified knowledge base. An example of this, is the discovery of a surprisingly large fraction of genes shared by eukaryotic genomes, which have a direct impact in DNA replication, transcription and metabolism.

The similarities of robotics and gene sequencing research are numerous. Both fields have shown enormous progress in specialized techniques and methodologies, but at the same time, they show a largely divergent terminology to describe similar concepts. Both need a 'common language' to describe devices, processes, and concepts, to allow for unifica-

tion and comparison between systems. This suggests that robotics can benefit from ontologies in the same way as genetics did in the past.

Robotic researchers have tried in the past to create robot ontologies for specific purposes like object categorisation [14] and robot navigation [3]. Few approaches however, have tackled the issue of building a robot ontology that focuses on the physical characteristics of robots. We argue that a "bottom-up" approach can be used to build a more general ontology that has application across fields. This approach would take the information stored in RoboDB and try to build the ontology based on factual data, e.g. the existence of legs, arms, the number of wheels available, etc. Relationships between components, sensors, and robots themselves would be based on the structural information (i.e. the connections) already specified in the embodiment descriptions.

We are aware that this method of ontology building would need a large number of robots before it can yield significant results. Even more, the ontology will constantly change as the knowledge of the field captured by RoboDB evolves. However, we believe that constructing such knowledge model will pay large dividends, as its applications are numerous.

5.4 Describing robot embodiments for robotic middleware

In Section 2 we suggested that modern robotic middleware can benefit from the semantic information gathered by RoboDB, as semantic annotations can be easily turned into alternative formats (e.g. RDF) easier to handle for computers.

The development of a common software architecture (i.e. a *robotic middleware*) has been one of the main topics of research in robotics during the last decades. It is generally recognized that it is hard to establish any kind of standards, due to the heterogeneous nature of the field. It has also been suggested that in order to design a common middleware, there must first be an agreement on a set of common abstractions and data types [20]. These standards are now becoming a necessity, as robotics research extends to areas as diverse in nature as psychology, and virtual worlds research [10]

Robotic middleware has approached the problem of embodiment description from different perspectives. Some middleware use configuration files containing a description of the sensors and actuators of the robot and their relationship (e.g. *Player/Stage* [9]). Others use an *Interface Definition Language (IDL)* as an intermediate representation to describe the robot's sensors and actuators and *services* (e.g. MIRO [23])

The robot descriptions produced by RoboDB can help to solve the problem of different knowledge representations used by robot middleware. A possible solution is the use of OWL-based documents for automatic code generation, as seen for example in [18]. Our idea is to transform the embodiment description stored in RoboDB to an OWL compliant format, then use this document to generate descriptions in one or more of the formats currently used by the most popular robot middleware. This can contribute to create a more homogeneous embodiment representation, and to lower considerably the level of expertise require to create such description. It would be no longer necessary to learn the intricacies and particularities of a specific middleware platform in order to add support for new hardware. Instead, a robot descrip-

tion could be created in RoboDB and then exported to IDL, Player/Stage, or other similar format.

We also foresee some of the difficulties we will face while pursuing this approach. For example, the flexibility RoboDB provides when describing a robot, can result in the level of detail in the description not being sufficient to generate a complete representation for the robot's capabilities. As a result some of the services the robot provides might not be available to the robot user. Therefore, the balance between the detail in the description and the semantic modeling required to generate such description is crucial to make it usable while keeping the complexity at manageable levels.

Finally, we acknowledge the need for further investigation of the shortcomings of SWT and their application to robotics. We intend to continue our research, focusing on improving the expressiveness of the robot description, on the possibility to build a robot ontology with this information, and on making its use within robotic middleware possible.

6. CONCLUSION

We presented a concept for the use of semantic web technologies to describe the physical structure of robotic embodiments. We showed examples of the need of an embodiment physical properties repository and the benefits to the HRI community. We introduced the prototype implementation of RoboDB, a SWT-based system that offers the functionality necessary to store meaningful information about the robot embodiment, by means of *semantic annotations*, and presented the heuristic evaluation of its interface. Finally, we discussed the possibilities of using the semantic information gathered to build and populate robot ontology, as well as its use as input for the automatic creation of embodiment descriptions that can be used by robot middleware.

Acknowledgment

7. REFERENCES

[1] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[2] D. Bakken. Middleware. *Encyclopedia of Distributed Computing*, 2002.

[3] J. Bateman and S. Farrar. Modelling models of robot navigation using formal spatial ontology. In C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, and Interaction*, volume 3343 of *Lecture Notes in Computer Science*, pages 366–389. Springer Berlin / Heidelberg, 2005.

[4] C. Burghart and R. Haeussling. Evaluation Criteria for Human Robot Interaction. *Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, page 23.

[5] A. Chella, M. Cossentino, R. Pirrone, and A. Ruisi. Modeling ontologies for robotic environments. In

Proceedings of the 14th international conference on Software engineering and knowledge engineering, page 80. ACM, 2002.

[6] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[7] J. Gellisen. Personal communication, July 7, 2010.

[8] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. CrubÈzy, H. Eriksson, N. F. Noy, and S. W. Tu. The evolution of protègè: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89 – 123, 2003.

[9] B. Gerkey, R. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, pages 317–323. Citeseer, 2003.

[10] A. Juarez, C. Bartneck, and L. Feijs. On the creation of standards for interactions between real robots and virtual worlds. *Journal of Virtual Worlds Research*, 2(3):4–8, October 2009.

[11] M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic mediawiki. In *The Semantic Web-ISWC 2006*, volume 4273/2006 of *Lecture Notes in Computer Science*, pages 935–942. Springer, Nov. 2006.

[12] B. Matthews. Semantic web technologies. *E-learning*, 6(6):8, 2005.

[13] S. McCullough. Force-directed graph layouts (`http://www.cricketschirping.com/weblog/2005/12/11/force-directed-graph-layout-with-proce55ing/`), Sept. 2010.

[14] R. Mendoza, B. Johnston, F. Yang, Z. Huang, X. Chen, and M. Williams. OBOC: Ontology Based Object Categorisation for Robots. In *Proceedings of the 4th International Conference on Computational Intelligence, Robotics and Automation (CIRAS 2007), Palmerston North, New Zealand*. Citeseer, 2007.

[15] R. Molich and J. Nielsen. Improving a human-computer dialogue. *Communications of the ACM*, 33(3):348, 1990.

[16] Moving Picture Experts Group (MPEG). MPEG-V working documents (`http://mpeg.chiariglione.org/working_documents.htm#MPEG-V`), Sept. 2010.

[17] F. Pfisterer, M. Nitsche, A. Jameson, and C. Barbu. User-Centered Design and Evaluation of Interface Enhancements to the Semantic MediaWiki. In *Workshop on Semantic Web User Interaction at CHI 2008*, 2008.

[18] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere. Towards an extensible context ontology for ambient intelligence. In P. Markopoulos, B. Eggen, E. Aarts, and J. L. Crowley, editors, *Ambient Intelligence*, volume 3295 of *Lecture Notes in Computer Science*, pages 148–159. Springer Berlin / Heidelberg, 2004.

[19] T. Quick, K. Dautenhahn, C. Nehaniv, and G. Roberts. On bots and bacteria: Ontology independent embodiment. *Advances in Artificial Life*, pages 339–343, 1999.

[20] W. Smart. Is a Common Middleware for Robotics Possible? In *Proceedings of the IROS 2007 workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware*. Citeseer, 2007.

[21] A.-Y. Turhan. Reasoning and explanation in $\mathcal{EL}$ and in expressive description logics. In U. Aßmann, A. Bartho, and C. Wende, editors, *Reasoning Web. Semantic Technologies for Software Engineering*, volume 6325 of *Lecture Notes in Computer Science*, pages 1–27. Springer Berlin / Heidelberg, 2010.

[22] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1):14–28, 2006.

[23] H. Utz, S. Sablatnog, S. Enderle, and G. Kraetzschmar. Miro-middleware for mobile robot applications. *IEEE Transactions on Robotics and Automation*, 18(4):493–497, 2002.

[24] H. Yanco and J. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pages 02–03, 2002.

[25] H. Yanco and J. Drury. Classifying human-robot interaction: an updated taxonomy. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2841 – 2846 vol.3, 2004.

[26] T. Ziemke. What's that thing called embodiment. In *Proceedings of the 25th Annual meeting of the Cognitive Science Society*, pages 1305–1310, 2003.