Proceedings of the 1st Standardized Knowledge Representation and Ontologies for Robotics and Automation, Workshop

Chicago, Illinois, USA

18th September, 2014

Hosted in

IEEE/RSJ International Conference on Intelligent Robots and Systems

Edited by Paulo J.S. Gonçalves, Craig Schlenoff, Edson Prestes, Tamás Haidegger

ISBN: 978-972-99143-9-3

http://www.est.ipcb.pt/laboratorios/robotica/iros-ora/

Organizing Committee

Workshop Co-chairs

- Paulo J.S. Gonçalves,
  LAETA/IDMEC, Instituto Superior Técnico, Universidade de Lisboa
  Polytechnic Institute of Castelo Branco, Portugal

- Craig Schlenoff,
  National Institute of Standards and Technology, USA

- Edson Prestes,
  Federal University of Rio Grande do Sul, Brazil

- Tamás Haidegger,
  Óbuda University, Hungary

Proceedings Production

- Paulo J.S. Gonçalves,
  LAETA/IDMEC, Instituto Superior Técnico, Universidade de Lisboa
  Polytechnic Institute of Castelo Branco, Portugal

International Program Committee

Howard Li (Univ. of New Brunswick, CANADA)
Maki Habib (AUC, EGYPT)
Moritz Tenorth (Technische Universität München, GERMANY)
K. Varadarajan (Vienna University of Technology, AUSTRIA)
Abdelghani Chibani (UPEC Univ., FRANCE)
Stephen Balakirsky (Georgia Tech, USA)
Alberto Vacarella (Politecnico Milano, ITALY)
Alessandro Saffiotti (Örebro University, Sweden)
Marcos E. Barreto (UFBA, Brazil)
Mara Abel (UFRGS, Brazil)

FOREWORD

Welcome to IEEE---ORA (Ontologies for Robotics and Automation) IROS workshop. This is the 1st edition of the workshop on Standardized Knowledge Representation and Ontologies for Robotics and Automation. The IEEE---ORA 2014 workshop was held on the 18th September, 2014 in Chicago, Illinois, USA.

In the IEEE---ORA IROS workshop, 10 contributions were presented from 7 countries in North and South America, Asia and Europe. The presentations took place in the afternoon, from 1:30 PM to 5:00 PM. The first session was dedicated to "Standards for Knowledge Representation in Robotics", where presentations were made from the IEEE working group standards for robotics and automation, and also from the ISO TC 184/SC2/WH7. The second session was dedicated to "Core and Application Ontologies", where presentations were made for core robotics ontologies, and also for industrial and robot---assisted surgery ontologies. Three posters were presented in emergent applications of ontologies in robotics.

We would like to express our thanks to all participants. First of all to the authors, whose quality work is the essence of this workshop. Next, to all the members of the international program committee, who helped us with their expertise and valuable time. We would also like to deeply thank the IEEE---IROS 2014 organizers for hosting this workshop.

Our deep gratitude goes to the IEEE Robotics and Automation Society, that sponsors the IEEE---ORA group activities, and also to the scientific organizations that kindly agreed to sponsor all the workshop authors work.

Paulo J.S. Gonçalves, Craig Schlenoff, Edson Prestes, Tamás Haidegger

WORKSHOP PROGRAM

1:30PM - Opening (5 min)

1:35PM - Session 1: Standards for Knowledge Representation in Robotics

Invited Speakers (60 min):

Craig Schlenoff, (NIST, USA | chair of the IEEE-ORA working group)

Title of presentation: “An Overview of the IEEE Ontology for Robotics and Automation Standardization Effort”

Francesco Amigoni (Politecnico di Milano, Italy), Wonpil Yu, (chair of the IEEE-MDR working group)

Title of presentation: “Standard for Robot Map Data Representation for Navigation”

Stephen Cameron (Professor, Department of Computer Science, Oxford University, UK) , Gurvinder Virk (Convener of ISO TC 184/ SC 2/WG 7 on Safety of personal care robots, and Convener of IEC SC 62A JWG9 Medical Electrical equipment and systems using robotic technology,

Title of presentation: “The ISO/IEC standardization efforts in robotics”

Edson Prestes (Federal University of Rio Grande do Sul (UFRGS), Brazil | co-chair of the IEEE-ORA working group)

Title of presentation: “Core Ontology for Robotics and Automation”

Panel discussion. (19 min) | Poster Teasers. (6 min)

3:00PM - Poster Session / Coffee break with refreshments

Zeynep Dogmus, Volkan Patoglu, Esra Erdem (Faculty of Engineering and Natural Sciences, Sabancı University, İstanbul, Turkey)

Title of presentation: “Ontological Query Answering about Rehabilitation Robotics”

Chris Paxton, Jonathan Bohren, Gregory D. Hager (Department of Computer Science, Johns Hopkins University, USA)

Title of presentation: “Towards a Standard for Grounding Symbols for Robotic Task Ontologies”

Ewerton Wantroba, Roseli Romero (Department of Computer Science, Institute of Mathematics and Computer Sciences, USP - University of São Paulo, Brazil)

Title of presentation: “A Method for designing Dialogue Systems by using Ontologies”

3:30PM - Session 2: Core and Application Ontologies

Invited Speakers (80 min):

Stephen Balakirsky (Georgia Tech, USA),

Title of presentation: “An Ontology for Industrial Robotics”

Paulo Gonçalves (LAETA/IDMEC, Instituto Superior Técnico, Universidade de Lisboa | Polytechnic Institute of Castelo Branco, Portugal),

Title of presentation: “An Ontology for Orthopedic Surgery”

Tamás Haidegger (Óbuda University, Hungary / ACMIT, Austria),

Title of presentation: “The Role of Ontologies in Robot-Assisted Surgery ”

4:50PM - Closing Remarks and Closing (10 min)

INDEX

An Overview of the IEEE Ontology for Robotics and Automation (ORA) Standardization Effort

Craig I. Schlenoff , *Senior* Member*, IEEE*

I. INTRODUCTION

One of the basic requirements for any type of robot communication (whether with other robots or humans) is the need for a common vocabulary along with clear and concise definitions. With the growing complexity of behaviors that robots are expected to perform as well as the need for multi-robot and human-robot collaboration, the need for a standard and well-defined knowledge representation is becoming more evident. The existence of such a standard knowledge representation will:

- more precisely define the concepts in the robot's knowledge representation;
- ensure common understanding among members of the community; and
- facilitate more efficient data integration and transfer of information among robotic systems.

The goal of the RAS Ontologies for Robotics and Automation Working Group (ORA WG) is to develop a standard to provide an overall ontology and associated methodology for knowledge representation and reasoning in robotics and automation, together with the representation of concepts in an initial set of application domains. It provides a unified way of representing knowledge and a common set of terms and definitions, allowing for unambiguous knowledge transfer among any group of human, robots, and other artificial systems.

The working group is composed of 157 members representing 23 countries and is made up of approximately 50% educational institutions, 25% private companies, and 25% government entities.

From the perspective of this working group, an ontology can be thought of as a knowledge representation approach that represents key concepts with their properties, relationships, rules and constraints. Whereas taxonomies usually provide only a set of vocabulary and a single type of relationship between terms (usually a parent/child type of relationship), an ontology provides a much richer set of relationships and also allows for constraints and rules to govern those relationships. In general, ontologies make all pertinent knowledge about a domain explicit and are represented in a computer-interpretable format that allows software to reason over that knowledge to infer additional information.

Craig Schlenoff is with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (phone: 301-975-3456; fax: 301-990-9688; e-mail: craig.schlneoff@nist.gov).

It would be extremely difficult to develop an ontology that could cover the entire space of robotics and automation. Hence, the working group is structured in such a way as to take a bottom-up and top-down approach to address this broad domain. This group is comprised of two subgroups entitled: Upper Ontology/Methodology (UpOM) and Industrial Robots.

II. UPPER ONTOLOGY/METHODOLOGY SUB-GROUP

The UpOM group has a goal to address high-level aspects for the ontology development, which include, but are not limited to:

- Defining the ontologies to be used;
- Coordinating the sub-ontologies built by other groups;
- Mediating the communication between the groups;
- Consolidating the sub-ontologies into a global ontology;
- Evaluating the ontology.

This group is reusing the concepts and formalisms of existing upper level ontologies to build the foundation of the upper ontology. The upper level ontology has high-level concepts that are linked to those used in the domain ontologies. In addition, this group provides support to make ontological decisions as transparent as possible in the resulting domain ontology. SUMO [1] is an example of a high-level ontology that was leveraged for this effort.

The activities developed by the sub-group, supported by the UpOM group, are based on the methodology proposed in METHONTOLOGY [2] and consist of the following phases:

1) Environment Study aims to acquire information on the platform to be used, the users, and applications where the ontology will be integrated;
2) Conceptualization provides a conceptual model related to the information acquired, including not only the concepts identified but also their relationships. When a model is mature, it will be submitted to the UpOM group to be reviewed;
3) Formalization and Implementation transforms the conceptual model into a computable model using OWL (Web Ontology Language);
4) Evaluate checks the consistency of the ontology;

5) Maintenance will be performed whenever new/altered/corrected knowledge is inserted into the ontology;
6) Documentation aims to produce a document that will be used to modify and/or reuse the ontology a posteriori.

The Industrial Robots subgroup is producing a subdomain ontology that will serve as a test case to validate the upper ontology and the methodology developed by Upper Ontology/Methodology subgroup. Once an initial version of the ontology is completed, it will be integrated into the overall ontology. During the integration process, as overlapping concepts are identified, a formal process will determine if these concepts should be merged, if they should be separated into two separate concepts, or if some other approach should be explored to reconcile them.

III. Industrial Robots Sub-Group

The Industrial Robots subgroup is developing an ontology focusing on the manufacturing kitting domain. This was chosen to align with an ongoing project at the National Institute of Standards and Technology (NIST) to enable robot agility in manufacturing kitting operations[1]. In order to maintain compatibility with the IEEE working group, the ontology has been fully defined in the Web Ontology Language (OWL) [3]. In addition, the ontology was also fully defined in the XML schema language [4].

The model has two top-level classes, SolidObject and DataThing, from which all other classes are derived. SolidObject models solid objects, things made of matter. DataThing models data. Each SolidObject A has at least one PhysicalLocation (the PrimaryLocation). A PhysicalLocation is defined by giving a reference SolidObject B and information saying how the position of A is related to B. Two types of location are required for the operation of the kitting workstation. Relative locations, specifically the knowledge that one SolidObject is in or on another, are needed to support making logical plans for building kits. Mathematically precise locations are needed to support robot motion.

The main classes of the ontology include:

- Kitting Workstation
- Changing Station
- Large Box With Empty Kit Trays
- Large Box With Kits
- Parts Tray With Parts
- Robot
- Stock Keeping Unit (SKU)
- Work Table

IV. Recent Efforts

The most recent efforts of the working group have been on the development of the core ontologies for robotics and automation (CORA) standard, which focuses on the representation of fundamental concepts from which the more detailed concepts belonging to other ORA WG ontologies are constructed.

The CORA ontology was proposed to assure an unambiguous definition of core notions of robotics and related topics. It is based on the ISO 8373:2012 standard, developed by the ISO/TC184/SC2 Working Group[2], which defines, in a natural language, terms that are important in the domain of Robotics and Automation (R&A). CORA goes beyond this standard by providing a set of formal (machine-processable) definitions and the relations amongst them. Moreover, concepts are defined for the robot itself and its interaction with the environment and/or other robots.

A consensus has been reached among the ORA working group participants, and the CORA proposed standard is expected to be forwarded to the IEEE Standards Association to start the balloting process in the October 2014.

If you are interested in getting involved in the ORA WG, please visit https://ieee-sa.centraldesktop.com/p1872public/ or contact the chair, Craig Schlenoff (craig.schlenoff@nist.gov) or vice chair, Edson Prestes (prestes@inf.ufrgs.br).

References

[1] A. Pease, I. Miles, and J. Li, "The Suggested Upper Merged Ontology: A Large Ontology for Semantic Web and its Applications," in *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, Edmonton, Canada, 2002.

[2] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "Methontology: from ontological art towards ontological engineering," 1997.

[3] D. L. McGuinness and F. Van Harmelen, "OWL web ontology language overview," *W3C recommendation,* vol. 10, p. 2004, 2004.

[4] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (XML)," *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210,* 1998.

[1] http://www.nist.gov/el/isd/ms/aprs.cfm

[2] http://www.iso.org/iso/iso_technical_committee?commid=54138

IEEE Standard for Robot Map Data Representation for Navigation

Wonpil Yu
Electronics and Telecommunications Research Institute
Daejeon, Korea
Email: ywp@etri.re.kr

Francesco Amigoni
Politecnico di Milano
Milano, Italy
Email: francesco.amigoni@polimi.it

***Abstract*—Robot navigation comprises three fundamental technologies to guide a mobile robot to its goal position: localization, mapping, and path planning. IEEE Map Data Representation (MDR) working group is developing a standard specification for representing a map used for robot navigation. The MDR standard aims to specify a common representation of two-dimensional maps for indoor and outdoor environments. This report describes the technical scope of the MDR standard and past activities of the MDR working group with respect to the MDR standard and current status thereof.**

I. Introduction

Autonomous robots, operating without guide tracks in uncontrolled environments are becoming common and economically viable. Robot navigation, which is an essential element for a mobile robot to be called an autonomous robot, may be defined as the process of determining and maintaining a course or a trajectory of a mobile robot to its goal location [1]. As shown in Figure 1, robot navigation comprises three fundamental technologies: localization, mapping, and path planning. As the figure shows, a robot needs some form of a map to perform a navigation task.

Complying with a standard for map data representation makes a vendor's components more compatible with others and therefore makes their products more desirable and more likely to win contracts. Standards compliance is particularly important in environments with devices from diverse vendors inter-operating, such as factories and military environments, where data interchange is a common occurrence. Being able to both use and provide the common data being shared amongst such devices is essential.

As can be expected, a common representation of robot environments brings a few advantages, including easy and economic exchange of maps, convenient benchmarking, facilitation of development and deployment of robotic applications, facilitation of technology development due to performance evaluation and technology exchange in terms of the common data format, and reduction of development and deployment costs.

The MDR working group has been approved by the IEEE Robotics & Automation Society (RAS) in July, 2011 to develop an international standard defining specifications of map data format. The goal of the MDR standard is to define a common map data representation (MDR) for robot navigation and an exchange interface for map data among robots, computers, and other devices.

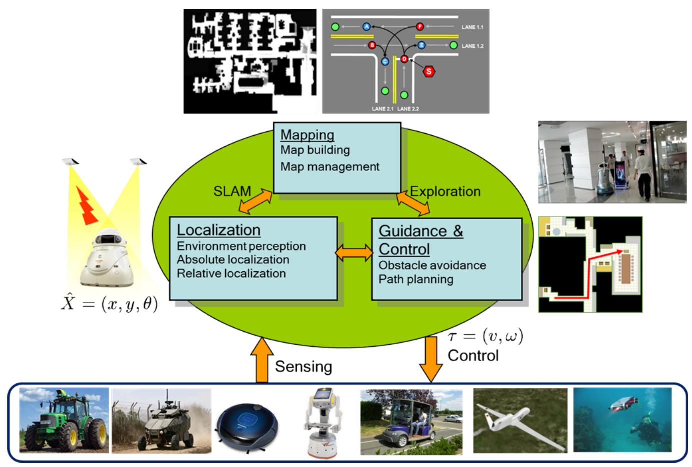

Fig. 1. The basic structure of robot navigation.

In what follows, the technical scope of the MDR standard is described in Section II. Section III describes formation of the MDR working group and policies and procedures defined in the MDR working group. Finally, Section IV provides the current status of MDR standardization activity and future actions to be made.

II. Technical Scope of the MDR Standard

Figure 2 shows a block diagram illustrating scenarios of a map creation use case. As shown in the figure, the map can be generated by a surveying tool, or given a priori in the form of a CAD file. Or the map can be generated by a mobile robot autonomously by using SLAM technology, for example. Like any other standards, the MDR standard does not limit the origin of the map, which is related to various kinds of mapping technologies; on the other hand, the MDR standard is concerned with encoding of spatial data so that exchange of maps can be carried out conveniently. Also, taking into account the industry practices and status of robot navigation technology, the MDR standard is concerned with the two-dimensional (2D) maps representing indoor and outdoor environments. No limit is imposed on the spatial complexity, geographic scale, or sensor modality.

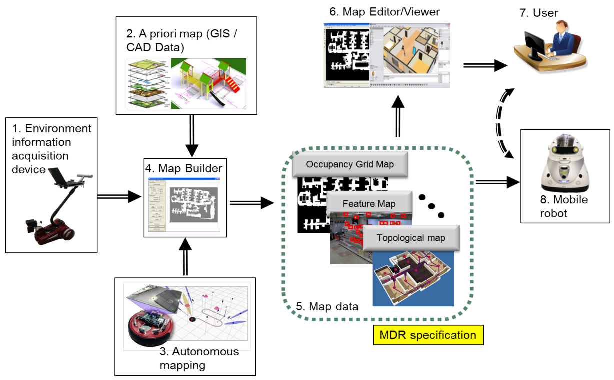

Fig. 2. The block diagram illustrating generation and use of a map for robot navigation.

In the MDR specification, we define a hierarchy of terminologies related to 2D robot maps for indoor and outdoor environments (see Figure 3). Based on the hierarchy of terminologies, the MDR standard defines a data model for each element of the hierarchy. To this purpose, a concept of a local map is introduced, which acts as an abstract base class representing a metric map or a topological map. LocalMap data type contains data types for map id, offset, map type, and coordinate system. Also, LocalMapTransform class is defined to deal with coordinate transformation between two different local maps. LocalMapType and CoordinateSystemType complete the LocalMap data type to describe a local metric or topological map. At this point, it should be noted that the MDR standard further defines pose uncertainty and coordinate transformation which are essential for mobile robotics but are not defined in other similar standards coming from ISO or OGC, for example.

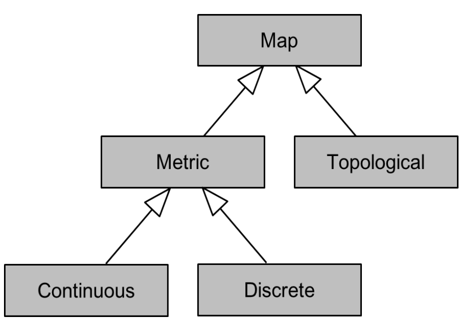

Fig. 3. The hierarchy of map classes defined in this standard.

Finally, the MDR standard specifies an XML format for map data exchange. To check validity (or conformance) of the map with respect to the MDR standard, an XML schema is defined. Implementation-specific extension of the base MDR specification includes definition of mdr_version, Metadata (meant for storing meta information related to a map, including author, email, license, map location, etc.) and MapArray is intended to store instances of LocalMaps.

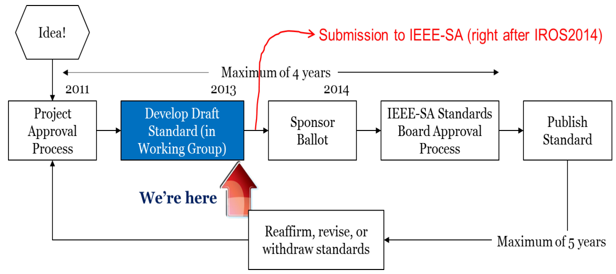

Fig. 4. IEEE-SA standards development process.

III. MDR Working Group

The Standing Committee for Standards Activities (SCSA) of the IEEE RAS is the technical sponsor of the MDR working group. The MDR working group submitted a project authorization request (PAR) to the IEEE-SA and obtains an approval for the project P1873 Standard for Robot Map Data Representation for Navigation" in November 2011. Since then the MDR working group members have been involved in developing the MDR standard, meeting two times a year at the conference weeks of ICRA and IROS. The MDR working group membership is automatically granted for those who attend the MDR meeting. The membership is granted for the participant attending at least one meeting of the MDR working group.

Currently, the MDR working group has two collaboration tools: Google group on IEEE RAS MAP Representation Standard and IEEE-SA central desktop on 1873 working group (https://ieee-sa.centraldesktop.com/1873workinggrouppublic/).

IV. Current Status and Further Actions

As of Sept. 2014, the standard abstract data model and concrete data format are almost defined. Some examples of APIs and protocols for translating map representations between the MDR standard and other formats are being developed. A graphical illustration of a current status of the MDR standardization is shown in Figure 4. The MDR standard draft is supposed to be submitted for a sponsor ballot around the end of 2014.

Once the MDR standard is successfully approved by the IEEE-SA, MDR standard for three-dimensional maps or semantic maps may be considered as the next item. As usual, selection of a new project will be determined by taking account of industry practices and needs.

References

[1] M. O. Franz and H. A. Mallot, "Biomimetic robot navigation," Robotics and AUtonomous Systems 30, pp. 133-153, 2000.

ISO-IEC standardization efforts in robotics

Gurvinder S. Virk, *Member, IEEE*, and Stephen Cameron, *Member, IEEE*

***Abstract*—The paper give an overview of robot standardization activities within ISO and IEC. Building from the early work of manufacturing robotics in industrial robotics, the work has extended to service robots comprising personal care and medical robotics. Although the emphasis of the work is on ensuring safety the standardization projects include the creation of a harmonized vocabulary for robotics and performance and inter-operability standards.**

***Index Terms*—ISO-IEC standardization, robot safety, robot types, industrial, service and medical robots.**

I. Introduction

THE importance of standardization as an enabling process in technical innovation is well recognized in helping to develop and maintain a global quality culture and even to set minimum requirement levels which are acceptable to the international market. The requirements presented in international standards allow manufacturers to sell their products everywhere without having to spend major effort modifying them for regional markets. On the other hand, they also act as a barrier against low quality or unsafe products unable to get certification and CE marking. In this respect safety standards are the most important.

Standards are voluntary, but they can be referred to, required by insurers, or incorporated in regulations (which are the mandatory rules and laws). There are a variety of standards resulting from international, regional, national, and even private initiatives and activities. They are generated and adopted for specific purposes, but international standards, such as those produced by ISO (International Organisation for Standardization; www.iso.org) or the IEC (International Electrotechnical Commission; www.iec.ch), have the greatest weight and are focused upon in this paper. ISO and IEC, in fact, consist of a network of national standards bodies, representing standardization activities in each country, from all regions of the world, working in partnership with international organizations such as the United Nations, their specialized agencies and the World Trade Organization. Significantly, ISO was founded in 1946 by delegates from 25 countries, began operating in 1947 and has now grown to 162 countries. The international importance of ISO standards are closely followed by IEC standards, which largely focus on electrotechnical issues. The IEC is also well supported globally with 82 member countries. The following types of standards (in priority rank) form the focus of international ISO-IEC standardization groups:

1. Safety standards
2. Vocabulary standards
3. Performance standards
4. Inter-operability (or modularity) standards.

Focusing now on robot standardization, until recently the only international robot safety standard which existed was ISO 10218 (Parts 1 and 2) applying to industrial robots and robot systems [1], [2]. Industrial robots have been traditionally designed to operate in isolation from humans (in real or virtual cages) and human-robot collaboration has been prohibited because of safety concerns. Recently there has been an interest to develop collaborative modes, but the lack of a safety standard for close human-robot interaction has been a problem. The situation has been the same for the new service robots; but since these are fundamentally designed to allow humans and robots to co-exist in the same space at the same time, the lack of international safety requirements has been a significant barrier.

In February 2014, ISO 13482 [3] was published as a harmonized safety standard for personal care robots aimed at applications involving close human-robot interaction as well as human-robot contact to provide a variety of services to humans for improving quality of life. This represents a major advance for the new service robots, but as it is a new standard, manufacturers and certification bodies need to become familiar with it so as to allow new products to enter the market with confidence. For this to happen, the research community has a vital role to play as most of the expertise for the new service robot domains lies with researchers.

The rest of the paper will present the different types of robot standards and details of the main active robot standardization work projects.

II. Types of Robot Types and Standards

Robotics has traditionally centered on industrial robots for manufacturing application since the early robots were developed in the 1960s but recently new robot products have been emerging to meet the growing demand for providing “services”. For specifying safety of robots, there are three main robot domains which present the general safety requirements for international commercialisation, namely:

- *Industrial robots*: these have high power, high precision

G. S Virk is Professor of Robotics with University of Gävle and KTH Royal Institute of Technology, Sweden, as well as Chairman and Trustee of CLAWAR Association Ltd, UK (e-mail: gurvinder.virk@hig.se).

S. A. Cameron is a Reader at the Department of Computer Science, University of Oxford, UK (e-mail: stephen.cameron@cs.ox.ac.uk).

defined tasks, designed to be used by skilled staff and traditionally designed on the "separate the robot from the human" principle in well-defined and controlled work-cell environments; they are the muster for powerful robots. This category is being extended to collaborative operation in semi-structured environments.

- *Personal care robots*: these are normally much lower power, do not need to be very precise, and are used by lay users in unstructured environments to perform a wide range of tasks most of which will involve close human-robot interaction and human-robot contact. These are home assistance robots.
- *Medical robots*: this is the muster for robots aimed at clinical applications, i.e., demonstrating performance better or equivalent to existing medical solutions, and being regulated under different rules from non-medical robots (namely, as medical devices rather than as machines).

Other robot sectors will benefit from these primary approaches for defining acceptable specific safety robot domain requirements such as driverless cars, unmanned flying robots, underwater robots, agriculture robots, etc. The current approach adopted in the development of international standards is to avoid the proliferation of standards for limited specific areas since the actual safety requirement would be very similar to each other in terms of causing harm and this can be prevented using "a combination of the safety requirements" in the three distinct domains described above.

III. ISO-IEC WORK PROJECTS ON ROBOT STANDARDIZATION

The main active robot standardization projects are the following:

- ISO TC184/SC2/WG1: Vocabulary and characteristics. Convener: Soon-Geul Lee, Korea; robot vocabulary, terms and definitions.
- ISO TC184/SC2/WG3 Industrial robot safety. Convener: Pat Davison, RIA, USA; safe human-robot collaboration in manufacturing applications.
- ISO TC184/SC2/WG7 Personal care robot safety. Convener: GS Virk, University Gävle/KTH/CLAWAR, SE/UK; non-medical service robot safety.
- ISO TC184/SC2/WG8 Service robots. Convener: Seungbin Moon, Korea; performance of service robots.
- ISO TC184/SC2/JWG9 Safety for medical electrical equipment and systems using robotic technology. Joint group with IEC TC62/SC62A. Convener: GS Virk; basic safety and essential performance of medical equipment with a degree of autonomy.
- ISO TC184/SC2/WG10 Modularity for service robots. Convener: GS Virk; Co-conveners, S Yang (China), S Park (Korea); hardware and software modular robot components.
- ISO TC199 SG on Safety data for human-machine interactions. Convener: Brian Tranter, CLAWAR, UK; normative safety related data for humans.
- IEEE Standard ontology for robotics and automation. Convener: Craig Schlenoff, NIST, USA
- IEC SC59F/WG5: Surface cleaning appliances. Convener: Sungsoo Rhim, Korea; cleaning performance of vacuum cleaning robots.

Of these the Modularity group (WG10) is the newest, and will have to consider how to combine different hardware and software components together in a manner that is 'safe' as well as functional.

IV. CONCLUSIONS

The paper has given an overview of how robotic standards are evolving globally and described the main ISO-IEC projects in robot standardisation. The focus on producing safety standards before products proliferate requires more input from the research community than occurs in the majority of other standards domains, but is seen as necessary given the general complexity of robotic systems.

ACKNOWLEDGMENT

The authors are grateful to the numerous international experts involved in the various robot standardization projects to assist in advancing the area of robotics. Prof. Cameron is also supported by EPSRC grant EP/J012564/1.

REFERENCES

[1] ISO 10218-1:2011 Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots. Harmonized standard.

[2] ISO 10218-2:2011 Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration. Harmonized standard

[3] ISO 13482:2014, Robots and robotic devices - Safety requirements for personal care robots. Harmonized standard.

Gurvinder S Virk (M'2012) is Professor of Robotics at University of Gävle, KTH Royal Institute of Technology, Sweden as well as being Chairman, Trustee of CLAWAR Association Ltd, UK. He is a technical expert in control theory and its applications with particular experience in service robotics and the use of advanced model-based control to a variety of applications. His current interests are in physical assistant exoskeletons for elderly persons (medical and non-medical applications), olfactory navigation for mobile robots, CLAWAR robots as well as integration of ICT into the built environment. He has extensive experience of project management and leading international R&D and standardisation projects.

Stephen Cameron (M'1990) is a Reader and Associate Professor at Oxford University. He has experience in many areas of robotics including planning, sensing, vision, and mobile vehicles. His current focus is on verification and on sensing and UAVs, and he sits on ISO working groups WG1, WG8, and WG10.

Core Ontology for Robotics and Automation

Edson Prestes and Sandro Rama Fiorini and Joel Carbonera

***Abstract*— This paper briefly presents the Core Ontology for Robotics and Automation (CORA), which was developed in the context of the IEEE Ontologies for Robotics and Automation Working Group. We discuss the importance of ontologies and standardization for the domain of Robotics and Automation and we present the main aims of this initiative. Furthermore, we present the results of an experiment that was conducted for testing the effectiveness of applying CORA for representing knowledge in a scenario that requires the cooperation of heterogeneous robots.**

I. Introduction

The advances of Robotics Systems around the globe show a very promising future where robots and humans will share the same environment as partners. Robots might work in factories, malls and airports. Controversially, they might even be used as lovers, as discussed by Levy [1]. In all cases, both human and robot should communicate to each other in order to exchange information. In this scenario, it is a *sine qua non* condition that humans and robots share a common vocabulary with clear and precise definitions of its terms. An effective way of achieving this goal is to standardize terminology in the Robotics and Automation (R&A) domain, as it has been done in the past in other domains [2], [3], [4]. A step forward from simple terminology is to ensure that agents share a common *conceptualization* (i.e. knowledge) about the world. In this paper, we briefly describe a standard for R&A that specifies domain concepts and relationships as an *ontology*.

Ontologies have gained popularity during the last years in Computer Science. They are information artifacts that formally represent shared conceptualization about a specific domain. They specify domain concepts and their meanings; together with concept relationships and constrains. Together, these constructs restrict properties and relationships to be assigned to entities in the domain. Often, ontologies are created as logical theories that restrict the meaning of a set of terms modeled as predicates. As such, ontologies are often used as base for *automated reasoning* and for ensuring the *semantic interoperability* in *information sharing* between computer systems.

This paper provides a brief overview of the ontology developed by the IEEE RAS Ontology for Robotics and Automation Working Group (IEEE RAS ORA WG). The ORA WG aims at developing a set of ontologies to standardize the knowledge representation in R&A. These ontologies will encompass different fields in R&A, including, but is not limited to, *Industrial Robotics*, *Surgical Robotics* and *Service Robotics*. To reach this goal, ORA WG developed a first set of ontologies covering some of the main concepts and relations in R&A.

*This work was supported by CNPq, CAPES and Petrobras PFRH-217

All authors are affiliated to the Informatics Institute, Federal University of Rio Grande do Sul (UFRGS), Brazil edson.prestes@ieee.org

II. Core Ontology for Robotics and Automation

In July, 2014, ORA WG submitted for approval the first standard draft to IEEE Standard Association (IEEE SA). The main contribution is the *Core Ontology for Robotics and Automation* (CORA). A *core ontology* encompasses terms that appear across the different subdomains of a larger domain. In the case of CORA, it includes concepts such as *robot*, *robotic system*, *robot part*, etc. The main role of CORA is to maintain the consistency among the different sub-ontologies in the standard. For instance, future ontologies about industrial and service robotics to be included in the standard shall commit to the the already existing concepts/definitions in CORA. Otherwise, inconsistencies and wrong inferences might appear.

CORA commits itself to the existence of four main notions in the whole domain of R&A: *device*, *robot*, *robot group* and *robotic system* (Figure 1). In brief, CORA defines robots as devices that exhibit agency (capability of acting by themselves), including devices ranging from clockwork robots to autonomous robots. Robots can form robot groups, which display their own agency. Finally, robots and robot groups, together with auxiliary equipment, can form robotic systems, such as robotic manufacturing system (see [5] for further discussion).

Three other ontologies support CORA: *CORAX*, *RPARTS* and *POS*. CORA was developed based on SUMO [6] (Figure 1); a top-level ontology that aims at defining the main ontological categories describing the world. However, SUMO does not cover every possible aspect of reality, even when we restrict ourselves to R&A. At the same time, some parts of reality are too general to be included in CORA. We introduced the CORAX ontology to address this problem by bridging parts of SUMO and CORA. In particular, CORAX includes concepts and relations associated with *design*, *interaction*, and *environment*, which are not covered in SUMO.

RPARTS is a sub-ontology of CORA that specifies the notions related to specific kinds of robot parts. This ontology also includes an ontological design pattern for modeling robot parts. According to CORA, robots are (agentive) devices composed of other devices. CORA considers that (i) it is not possible to determine in advance what kinds of devices can or cannot be robot parts; and (ii), none of the instances that can be classified as robot parts are essentially robot parts, since they can exist by themselves when they

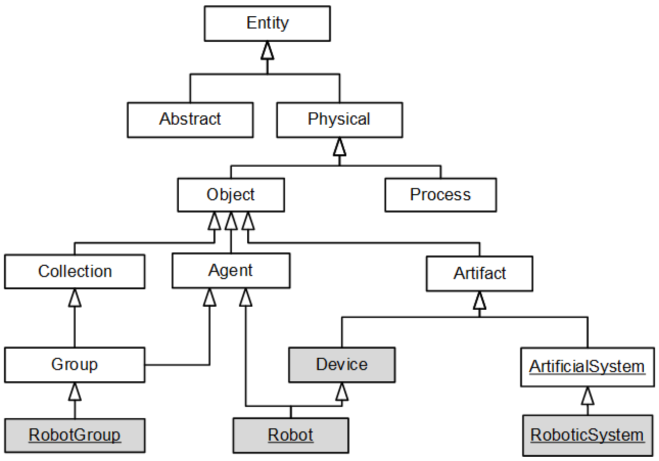

Fig. 1. Overview of the taxonomy of SUMO and the main CORA concepts (underlined). The concept *Device* is a concept of SUMO.

are not connected to a robot (or when they are connected to other complex devices). Due to this characterization, CORA defines *robot part* as a *role* played by devices while they are connected to the robot. According to CORA, robot parts can be: *robot sensing parts*, *robot actuating parts*, *robot communicating parts* and *robot processing parts*.

POS is an ontology that extends SUMO and complements CORA, by specifying the main concepts and relations underlying the notions of *position*, *orientation* and *pose*. POS allows to describe positions, orientations and poses in *coordinate systems*, which are *abstract* entities that are defined in relation to a single *reference object*. This ontology also provides means of describing positions and orientations *qualitatively* and *quantitatively*. Regarding the information about position, in the quantitative case, it can be described as a *position point* in a given coordinate system; and, in the qualitative case, it can be described as a *position region* defined as a function of a reference object. Besides these concepts, POS also specifies other important notions such as *transformation* and *spatial operator*. A detailed discussion on POS, CORAX and RPARTS can be found in [7], [8].

III. CORA Development

The development of CORA follows the METHONTOLOGY [9] methodology. METHONTOLOGY is a mature ontology engineering methodology for building ontologies, specifying a set of activities and techniques to help the ontology development process. The ORA WG has begun its work by defining the scope of the main ontologies that should compose the standard. The necessity of a core ontology was clear from the beginning, given the vast extent of the area and the necessity to define some basic notions that are common to all subareas of R&A. Once the goal of developing CORA was established, ORA WG started specifying the sources from which the domain knowledge could be acquired. The sources that were identified include existing standards in the domain, textbooks, peer-reviewed papers, domain experts and other ontologies in the domain. The main selected source was the ISO/FDIS 8373 document, elaborated by ISO/TC 182/SC 2.It provides generic terms defined in natural language, which are common across the R&A domains (such as robot, joint, actuators, etc). The ISO/FDIS 8373 standard served as a general direction to the scope of CORA. Based on terms and definitions from ISO/FDIS 8373 and knowledge acquired from other sources, we specified a first taxonomy of concepts and relationships. Later on, we introduced SUMO in the development, which helped to further enhance the ontological commitment of the entities in our ontology. Other valuable source of knowledge for our project was the *Autonomy Levels For Unmanned Systems (ALFUS)* framework [10], which presents an extensive study on autonomy in unmanned vehicles. In short, ALFUS states that autonomy is generally dependent on the degree of *human intervention* and *context*, where the latter is characterized by type of mission and environment. CORA's definition of autonomy is closely related to what ALFUS defines *modes of operation* for unmanned systems. These modes stretch from *remote controlled* to *fully autonomous*, representing the degrees of human interaction needed for the robot to perform its task. After the main concepts of CORA have been identified and formalized, we implemented the ontology in SUO-KIF language [6]. This process of knowledge acquisition/formalization/implementation was repeated a number of times until the group of experts participating ORA WG agreed on the entities and definitions in CORA. The other sub-ontologies, such as CORAX, POS, etc., are sub-products of that process.

IV. Preliminary Results

Besides the recent submission of CORA for IEEE SA appreciation, some current initiatives within the ORA WG have been focused on developing scenarios for testing the effectiveness of CORA for knowledge representation in R&A applications. In [11], Jorge et al. developed a scenario comprising the use of CORA in a task involving *human-robot* and *robot-robot interactions*. This scenario has three different actors: a *human*; a humanoid NAO H25, called *manipulator*; and a Pioneer 3DX, called *transporter*. The goal is to explore the use of CORA for *mediating the interactions* among *heterogeneous agents*, with different sensors and different capabilities. In this setting, all communication among these actors is done through a centralized system, called *ontology server*.

In this scenario, the human requests a cargo (i.e., a pen) to the server. The system checks which agent has the cargo and send a message to the holder. The holder, which is the manipulator, informs the system about this. The transporter robot moves in the environment towards the manipulator and both align themselves to allow the manipulator to drop the cargo on the top of transporter. The manipulator then informs the server that it has dropped the cargo. The transporter get aware about this fact and moves towards the human. When transporter aligns itself to the human position, the human picks up the cargo. All the information shared among the agents (such as positions, orientations and references to other agents) through the ontology server was specified in accordance to CORA. The task is very simple, however, it is used as a proof of concept about the use of CORA in a real scenario. For more details, we refer the reader to [11]. Currently, ORA members are developing other applications to an industrial scenario. In [8] it is discussed possible tasks and scenarios in R&A that can take advantage of CORA.

V. Conclusion

This paper presents some basic notions and motivations behind CORA development. We see CORA as an step forward in the necessary standardization of R&A domain given the clear ontological commitments it provides. Also, it is important to note that CORA is the first standard developed within IEEE Robotics and Automation Society. We hope that, once it is approved, CORA is used by all stakeholders in the development and use of robotic systems.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support from CNPq, CAPES and Petrobras PFRH-217.

References

[1] D. Levy, *Love and sex with robots: the evolution of human-robot relationships*. Harper Perennial, 2008.

[2] I. of Electrical and E. E. IEEE, "Graphic symbols for electrical and electronics diagrams," 1975.

[3] ——, "Ieee standard for ethernet," 2012.

[4] ——, "Ieee standard for system and software verification and validation," 2012.

[5] E. Prestes, J. L. Carbonera, S. Rama Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Gonçalves, M. E. Barreto, M. Habib, A. Chibani, S. Gérard, Y. Amirat, and C. Schlenoff, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193–1204, 2013.

[6] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, ser. FOIS '01. New York, NY, USA: ACM, 2001, pp. 2–9.

[7] J. L. Carbonera, S. R. Fiorini, E. Prestes, V. A. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, T. Haidegger, M. E. Barreto, *et al.*, "Defining positioning in a core ontology for robotics," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1867–1872.

[8] S. Rama Fiorini, J. L. Carbonera, P. Gonçalves, V. A. M. Jorge, V. F. Rey, T. Haidegger, M. Abel, S. A. Redfield, S. Balakirsky, V. Ragavan, H. Li, C. Schlenoff, and E. Prestes, "Extensions to the core ontology for robotics and automation," *Accepted for publication in Robotics and Computer-Integrated Manufacturing*, 2014.

[9] M. Fernández, A. Gómez-Pérez, and N. Juristo, "METHONTOLOGY: from ontological art towards ontological engineering," in *Ontological Engineering*, ser. AAAI Spring Symposium, vol. 6, 1997, AAAI Spring Symposium, pp. 33–40.

[10] H. M. Huang, K. Pavek, B. Novak, J. Albus, and E. Messina, "A framework for autonomy levels for unmanned systems (ALFUS)," in *Proc. of the 2005 AUVSI's Unmanned Systems North America Symposium, Baltimore, MD*, 2005, pp. 1–9.

[11] V. A. M. Jorge, V. F. Rey, R. Maffei, S. Rama Fiorini, J. L. Carbonera, F. Branchi, J. a. P. Meireles, G. S. Franco, F. Farina, T. S. d. Silva, M. Kolberg, M. Abel, and E. Prestes, "Exploring the ieee ontology for robotics and automation for heterogeneous agent interaction," *Accepted for publication in Robotics and Computer-Integrated Manufacturing*, 2014.

Implementation of an Ontology for Industrial Robotics

Stephen Balakirsky and Andrew Price

***Abstract*— The Industrial Subgroup of the IEEE Robotics and Automation Society's Ontologies for Robotics and Automation Working Group has developed extensions to the group's core ontology to support assembly tasks in industrial settings. At this time, the ontology has focused on kit building, which is a simple, but relevant subdomain of assembly. This paper focuses on an implemented system that is designed to take advantage of the IEEE ontology in order to provide flexibility and agility to the kit building process. The architecture of the system as well as the interaction between the running system and the ontology are discussed. This includes both how real-world knowledge is entered into the ontology as well as how that knowledge is extracted and utilized by the system.**

I. Introduction

Many of today's robotic arms are capable of obtaining sub-millimeter accuracy and repeatability. Robots such as the Fanuc LR Mate 200*i*D claim $\pm$ 0.02 mm repeatability [11] which has been verified in various publicly viewable experiments [3] [7]. However, these same systems lack the sensors and processing necessary to provide a representation of the workcell in which they reside or of the parts that they are working with. In fact, according to the International Federation of Robotics (IFR), over 95% of all robots in use do not have a sensor in the outer feedback loop. They rely on fixtures to allow them to be robust in the presence of uncertainty [4]. This lack of sensing in the outer feedback loop leads to systems that are taught or programmed to provide specific patterns of motion in structured workcells over long production runs. These systems are unable to detect that environmental changes have occurred, and are therefore unable to modify their behavior to provide continued correct operation.

Just-in-time manufacturing and small batch processing requires changes in the manufacturing process on a batch-by-batch or item-by-item basis. This leads to a reduction in the number of cycles that a particular pattern of motion is useful and increases the percentage of time necessary for robot teaching or programming over actual cell operation. This teaching/programming time requires that the cell be taken off-line which greatly impacts productivity. For small batch processors or other customers who must frequently change their line configuration, this frequent downtime and lack of adaptability may be unacceptable.

Research aimed at increasing a robot's knowledge and intelligence has been performed to address some of these issues. It is anticipated that proper application of this intelligence will lead to more agile and flexible robotic systems. Both Huckaby et al. [8] and Pfrommer et al. [10] have examined the enumeration of basic robotic skills that may be dynamically combined to achieve production goals. The EU-funded RObot control for Skilled ExecuTion of Tasks in natural interaction with humans (ROSETTA) [9] and Skill-Based Inspection and Assembly for Reconfigurable Automation Systems (SIARAS) [13] have proposed distributed knowledge stores that contain representations of robotic knowledge and skills. The focus of these programs is to simplify interaction between the user and the robotized automation system.

The IEEE Robotics and Automation Society's Ontologies for Robotics and Automation Working Group (ORA) [12] has also taken the first steps in creating a knowledge repository that will allow greater intelligence to be resident on robotic platforms. The Industrial Subgroup of this working group has applied this infrastructure to create a sample kit building ontology. Kit building may be viewed as a simple, but relevant manufacturing process.

Balakirsky et al. [1] describes the Industrial Subgroup's ontology, the knowledge that it contains, and a Planning Domain Definition Language (PDDL) [6] planning system that is able to dynamically alter the system's operation in order to adapt to variations in its anticipated work flow. The system does not require *a priori* information on part locations (i.e fixturing is not required) and is able to build new kit varieties without altering the robot's programming. While this body of work presents the high-level concepts behind the workcell, the details of how the ontology is actually utilized by the system are absent. This paper aims at addressing this omission by presenting the knowledge flow from the detected real-world objects into the ontology, and then from the information contained in the ontology into an executor that is able to convert high-level abstract commands into actions that are bound to specific instances of objects in the world.

This architecture for the system (shown in Figure 1) depicts the classic sense, model, act paradigm that is followed by the workcell. All of the workcell's activities are controlled by the *Executor* which has the responsibilities of validating each proposed high-level abstract plan step contained in a Problem Definition Domain Language (PDDL) precomputed plan file, binding abstract variables contained in the plan step to actual real-world instances, commanding actions from the *Robot Controller*, and verifying that actions were successful. The *Executor* is also able to dictate a specific focus of attention to the sensing system in order to assure that the world model is accurate and relevant to operations being performed.

Stephen Balakirsky and Andrew Price are with the Georgia Tech Research Institute, Atlanta, GA, USA. Stephen.Balakirsky@gtri.gatech.edu, Andrew.Price@gtri.gatech.edu

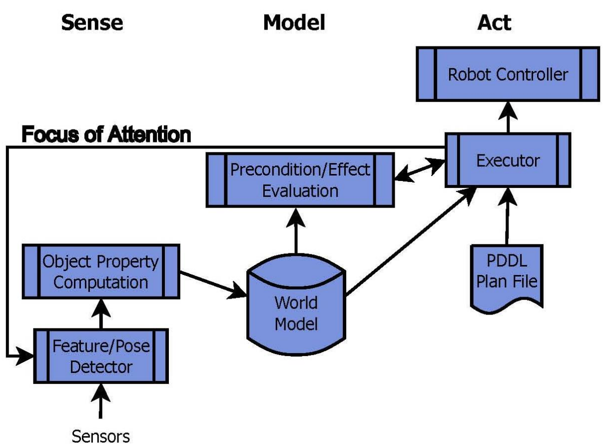

Fig. 1. The system follows a classic sense, model, act paradigm with the world model providing information for instance binding.

The organization of the remainder of this paper is as follows. Section II provides an overview of the PDDL planning language that is utilized to represent high-level abstract plans. Section III discusses the detailed operation of binding specific instances to the abstract plan, Section IV discusses how PDDL and the ontology are combined in order to provide verification and validation of actions, and Section V discusses how simple vision processing techniques are utilized to populate the ontology. Finally, Section VI presents conclusions and future work.

II. PDDL

The objective behind domain independent planning is to formulate a planner that is able to construct plans across a wide variety of planning domains with no change to the planning algorithms. The typical problem presented to such a planner consists of:

- A set of objects,
- A set of predicate expressions that define properties of objects and relations between objects,
- A set of actions that are able to manipulate the predicate expressions,
- A set of predicate expressions that constitute a partial world state and make up the initial conditions,
- A set of problem goals, which are predicate expressions that are required to be true at the end of a plan.

Several of the items listed above rely on *predicate expressions*. A sample *predicate expression* is shown in Figure 2. *predicate expressions* may be defined as binary expressions that contain one or two grounded objects or instances as arguments and evaluates to a truth value. Individual facts may be proven by evaluating a *predicate expression*, or groupings of *predicate expressions* that evaluate to a given truth value may be used to provide a partial definition of the world's state. For example, the *predicate expresion* shown in Figure 2 evaluates to *true* if the tooling in use by the robot is the correct tooling for manipulating a part of the given Stock

```
Sample predicate expression:
(tooling-is-for-SKU ?tooling ?sku)

Sample action:
(:action take-part
 :parameters(
  ?robot - Robot
  ?sku - StockKeepingUnit
  ?tray - Tray
  ?tooling - Tooling)
 :precondition(and
  (> (num-parts-in-tray ?tray) 0.000000)
  (tooling-is-for-SKU ?tooling ?sku)
  (tooling-empty ?tooling)
 :effect(and
  (decrease (num-parts-in-tray ?tray) 1)
  (not(tooling-empty ?tooling))
  (part-loc-ref-tooling ?part ?tooling)
  (not(part-loc-ref-tray ?part ?tray))
)
```

Fig. 2. Sample PDDL syntax for predicate expressions and action definitions.

Keeping Unit (SKU) type. Note that this predicate is true if the tooling works for a given type or class of part and does not bind the plan to a particular instance of such a part.

Predicate expressions may be used for *preconditions* (expressions that must be true for an action to be executed) as well as *effects* (expressions that are expected to become true as the result of an action). If an *action*, as shown in the bottom half of Figure 2, is defined as a fully-instantiated operator, then the job of the planner is to formulate a sequential list of valid actions, referred to as a *valid plan*, which will cause the set of predicate expressions that represent the initial state to transition to the set of predicate expressions that satisfies the problem's goals.

III. Instance Binding

As shown in the architecture of Figure 1, the *Executor* reads a plan file that contains a series of abstract PDDL actions and works with the system's *World Model* to formulate concrete commands to be executed by the *Robot Controller*. The *World Model* contains tables that have been automatically constructed from the ontology, and populated by the sensor processing system.

As shown in Figure 2, each PDDL action contains a set of parameters that are of specific types. These types correspond to classes in the IEEE RAS ORA Industrial Ontology, and the instances of these classes are represented in the *World Model*. The class relationships from the ontology combined with the properties of actual instances may be utilized to bind the action to an actual instance of a class and to extract the information necessary to send concrete actions to the *Robot Controller*.

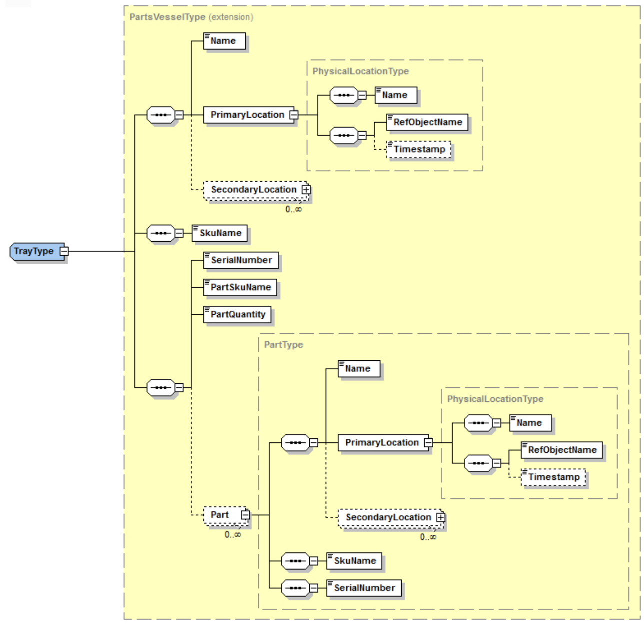

Fig. 3. Depiction of the *Tray* class from the ontology. This class contains information on the location of the tray, the general class of parts contained in the tray, and the locations of specific part instances.

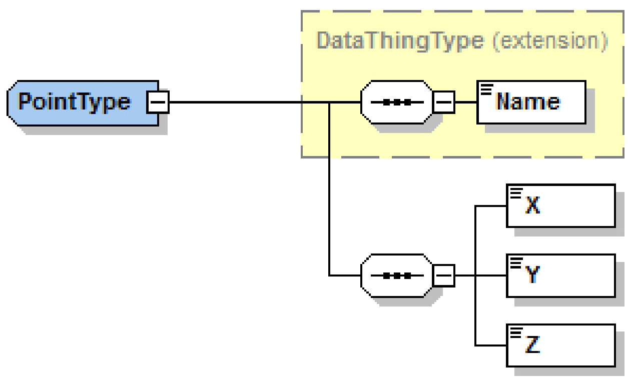

Fig. 4. Depiction of the *PointType* class from the ontology. This is a Level 0 class that has all of its properties as fundamental types (all floating point numbers). The class contains information on the spatial location of a point in 3-dimensional space.

A. World Model Construction

Classes in the ontology may extend other classes from the ontology as well as contain properties that are also classes. This may be seen in Figure 3 where the class *TrayType* inherits from the class *PartsVesselType* and contains 0 or more instances of *PartType* as well as a *PhysicalLocationType*. Since these types are not fundamental types, they may be further decomposed into additional types. For example, the *PhysicalLocationType* is implemented as a *PoseLocationType* which contains one *PointType* and two *VectorTypes*. Classes that only contain properties that project to fundamental types are known as *Level 0* classes since no further projections are possible. Once such class is the *PointType* class and is depicted in Figure 4.

Each class from the ontology is given its own data table in the *World Model*. However, in order to assure that the database is in 3^{rd} normal form, only a class's properties that project to fundamental elements and references to contained classes (and their table entries) are included in the columns of the table. Only Level 0 classes contain no columns which reference other tables. Individual instances of each class become the rows of the table. 3^{rd} normal form prevents duplicate information in the database and assures that it is possible to search over all of the information,

B. Example of Instance Binding

This subsection will provide an example of how the ontology and *World Model* are utilized to perform one of the main tasks in building a kit. That task is retrieving a part from a part's tray to place in the kit. As shown in Figure 2,this action is referred to as *take-part* and it relies on the classes of *StockKeepingUnit*, *Tray*, and *Tooling* from the ontology. A representation of the *Tray* class may be seen in Figure 3. In order to take a part from the tray, the system must first find a specific part instance that exists in the tray and must then find that part's location in world coordinates so that the robot may be commanded to move to the correct position for picking the part.

The fist step in this process is to identify a particular instance of the part for retrieval. The existence of such a part may be verified by checking the *PartQuantity* property of the part tray. If this value is greater than zero, a query may be performed on the *Part*s table to return all parts that are both of the SKU being sought and have a reference pointer which points to the instance of the part tray.

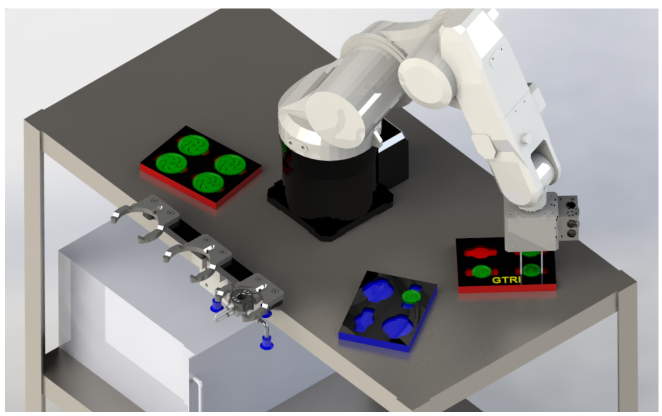

Fig. 5. A rendering of workcell. The robot is equipped with a tool changer and is tasked with building a kit from various trays of parts. No fixtures are utilized for part placement, which may occur at any point within the reach-space of the robot.

Reference pointers are utilized to allow local measurements of item locations to be performed as well as to allow for reasoning in the ontology. Figure 5 depicts our typical workcell arrangement while Figure 6 displays the hierarchy of coordinate frames. With this arrangement, a particular part instance would be identified from the tray, and this part would be located in the *World Model* with respect to

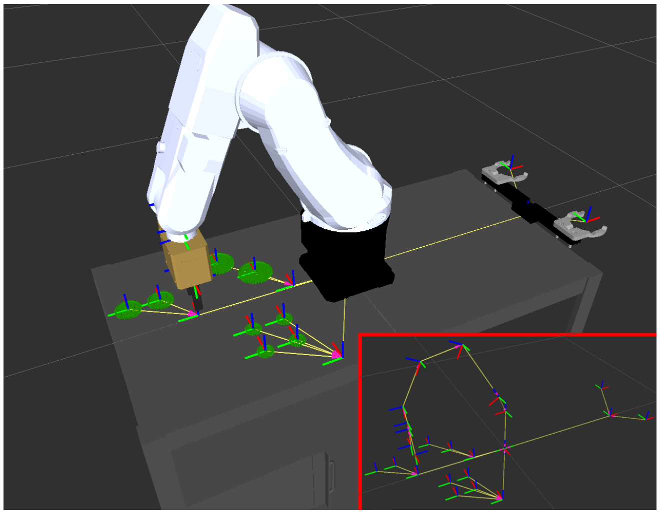

Fig. 6. A rendering of the coordinate system hierarchy that is specified in the ontology. The tri-color shapes represent the coordinate frames and the yellow lines are the reference pointers to parent frames. The inset region shows the underlying tree structure, while the main window shows the database objects in the workspace.

the tray. In turn, the tray would be located with respect to the table, and the table would be located with respect to the workcell itself. This workcell frame acts as the global coordinate system. By following this chain of references, the global location of the part may be determined.

The location of the part is one critical piece of information that is necessary for picking the particular part instance. Other information that is required includes the gripper that should be used to pick the part, and the precise grasp locations that are available for part gripping. Once again, all of this information is described in the ontology and available for the *Executor* to use through tables in the *World Model*.

IV. Action Verification and Validation

While world model information is used to ground movements of the robot to specific object instances, it is also utilized to validate that command execution is possible and to verify that commands were successful. As shown in Figure 2, each command contains a block of *precondition*s and a block of *effect*s. These blocks are composed of *Predicate Expressions* that may be evaluated directly from information in the *World Model* database.

For the *precondition*s, the predicate expressions may be examined and verified directly from the information that is encoded in the ontology. For example, the *precondition* shown in Figure 2 that refers to the number of parts in the tray may be directly verified from the *PartsVessleType* table.

For the *Effect*s, the information described in the *Predicate Expressions* is used to set the focus of attention of the sensor processing system so that the various items may be verified through inspection. For example, the *Predicate Expression* shown in Figure 2 that refers to the tooling not be empty is designed to verify that a successful grasp occurred. This may be verified by directing the sensor processing system to verify that the tooling is indeed not empty.

V. Instance Updates

The *World Model* database is grounded through the use of an external vision system. This subsystem is responsible for pushing updates to the planning and execution systems by way of the database, and is also responsible for action verification. After rectifying and balancing an incoming image from a tripod-mounted camera, the computer vision system leverages the following techniques to push updates to the database server.

- For this experimental setup, surrogate parts trays are created by printing part outlines and fiducial markers onto paper sheets. Each tray is represented in the ontology as a collection of slots and fiducial tags. The position and content of each tag is stored in the system, and the whole collection of tags for a given tray is tracked as a rigid body, providing more stable position and orientation data, as well as robustness to occlusions (a frequent occurrence when the robot is manipulating parts). Figure 7 shows the 6DOF location of the origin of parts and kit trays as 3D axes projected into the camera view. Image processing for this component is provided by the ALVAR library [15].
- In the current system, parts can be detected (but not identified) by means of elementary image processing techniques, including HSV-space thresholding, basic morphological operations, and connected components determination. In Figure 7, detected parts are shown with red bounding boxes. These tools are easily available in the OpenCV library [2]. Interested readers should refer to Chapters 2.3 and 3.3 of [14] for more information.
- Gears can be identified via keypoint extraction and histogram clustering. A sub-window is computed from the part detection mask, and SURF features are calculated over the region. These features are clustered and histogrammed to create a single feature for the image window, following the typical Bag-of-Words classification approach [5]. Finally, a set of Support Vector Machine classifiers, pre-trained on a small (c. 200 hand-labeled images) dataset, is used to categorize the detection region as one of the model classes. Again, these features are available through OpenCV. For a future implementation, a model-based wireframe tracking system is being investigated to perform both matching and tracking.

Once the location of a particular gear instance is identified, it may be updated in the database. At this time, the reference pointer to a parent object is updated not through the vision system, but through the effects clauses of the PDDL action. In other words, the vision system only updates object positions while respecting the existing reference frame hierarchy's topology.

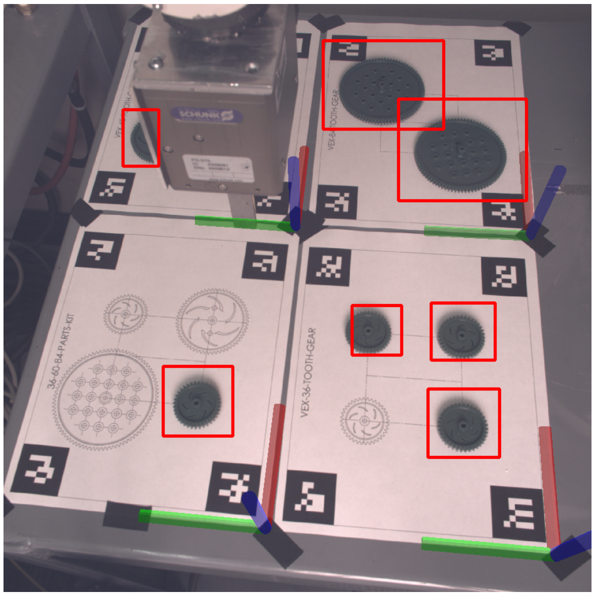

Fig. 7. Vision subsystem identifying and locating parts trays and present gears.

VI. Future Work

While we have successfully put together all of the individual pieces of this system, we have not yet run a fully closed loop demonstration. This represents our short-term goal for the project where a user will be able to place trays of parts and a kit anywhere within the reach-space of the robot and the robot will assemble the correct kit.

Improvements are also planned for each of the individual subsystems ranging from the automatic determination of part-to-part relationships (the reference pointers of the parts), to enhancing the database and *Executor* to take advantage of the fact that the PDDL commands are also represented in the ontology.

Acknowledgments

This work is partially supported by the National Institute of Standards and Technology under cooperative agreement 70NANB14H025. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the author and should not be interpreted as representing the official policies, either expressly or implied, of the funding agencies of the U.S. Government.

References

[1] S. Balakirsky, Z. Kootbally, T. Kramer, A. Pietromartire, C. Schlenoff, and S. Gupta. Knowledge driven robotics for kitting applications. *Robotics and Autonomous Systems*, pages 1205 – 1214, 2013.

[2] G Bradski. The OpenCV Library. *Dr Dobbs Journal of Software Tools*, 25(11):120–125, 2000.

[3] Control and Montreal Robotics Lab at the ETS. Measuring the absolute accuracy of an abb irb 1600 industrial robot. *http://www.youtube.com/watch?v=d3fCkS5xFlg*, 2011.

[4] Statistical Department. World robotics industrial robotics. Technical report, International Federation of Robotics, 2012.

[5] Li Fei-Fei and Pietro Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, 2005.

[6] M. Ghallab, A. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl–the planning domain definition language. Technical Report CVC TR98-003/DCS TR-1165, Yale, 1998.

[7] Olli Hnninen. Repeatability test fanuc robot. In *https://www.youtube.com/watch?v=8DMT5Aj_jjc*, 2014.

[8] Jacob Huckaby and Henrik I Christensen. A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics. In *Workshops at 26th AAAI Conference on Artificial Intelligence*, 2012.

[9] Rajendra Patel, Mikael Hedelind, and Pablo Lozan-Villegas. Enabling robots in small-part assembly lines: The" rosetta approach"-an industrial perspective. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–5. VDE, 2012.

[10] Julius Pfrommer, Miriam Schleipen, and Jurgen Beyerer. Pprs: Production skills and their relation to product, process, and resource. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–4. IEEE, 2013.

[11] Fanuc Robot. Fanuc robot lr mate 200id. In *Data Sheet*, 2013.

[12] Craig Schlenoff, Edson Prestes, Raj Madhavan, Paulo Goncalves, Howard Li, Stephen Balakirsky, Thomas Kramer, and Emilio Miguelanez. An ieee standard ontology for robotics and automation. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1337–1342. IEEE, 2012.

[13] Maj Stenmark, Jacek Malec, Klas Nilsson, and Anders Robertsson. On distributed knowledge bases for small-batch assembly. In *Cloud Robotics Workshop, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[14] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[15] VTT Technical Research Centre of Finland. Augmented Reality / 3D Tracking — ALVAR. www.vtt.fi/multimedia/alvar/. Accessed: 2014-20-08.

An Ontology for Orthopedic Surgery

Paulo J. S. Gonçalves

***Abstract*— This paper presents the current developments of an ontology for orthopedic surgery (OROSU), based on the Core Ontology for Robotics and Automation (CORA). CORA was developed by the Ontologies for Robotics and Automation Working Group, under the scope of the IEEE Robotics and Automation Society. OROSU is developed based on the biomedical ontologies that already exist, and proposes a framework to represent robotic and orthopedic surgery knowledge, to be used directly by humans (surgeons, nurses, technicians, and so on), working with robots. Examples are shown of the OROSU, applied to Hip Surgery surgical procedures.**

I. Introduction

The biomedical field assisted, in the past decades, significant efforts towards standardization of: data, vocabularies, surgical procedures, and so on. These efforts, are really challenging and undertake a tedious work, due to the huge amount of specialities in the biomedical field. This fact is also an important issue when we focus on the orthopedic field [2].

In surgery, several goals must be achieved, e.g., safety, efficiency, and nowadays a cost effective solution should be seek. These goals led to the introduction of robots in surgery [1]. Today, these machines, teleoperated by humans, can help in navigation, and can also reduce the surgeon hand tremor. With the use of robots in surgery, less invasive, more precise, and cleaner surgical procedures, can be achieved.

This paper is focused on the presentation of a knowledge model for orthopedic robotic surgery, using ontologies. From this narrow field, a bottom up approach can be used to obtain the knowledge from both, medical and robotic fields. In other words, the ontology for orthopedic robotic surgery (OROSU) must be obtained from ontologies and standards on related fields.

This paper is organized as follows. Section II, presents the Ontology for Orthopedic Surgery. Section III, presents examples of reasoning results, based on the ontology. The paper ends with section IV, where some conclusions are drawn.

II. The Ontology for Orthopedic Surgery

Conceptual knowledge was modelled using ontologies, in [7], with special focus on Computer Aided Surgery. There, the authors developed Surgical Ontologies for Computer Assisted Surgery (SOCAS), based on the General Formal Ontology (GFO) [6]. Current research on this subject aims to apply such models in the operating room, that allows the integration of robotic ontologies during surgery.

Paulo J. S. Gonçalves, is with LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, and Polytechnic Institute of Castelo Branco, Portugal. paulo.goncalves@ipcb.pt

Robotics development is pushing robots to closely interact with humans on real world unconstrained scenarios, that surely complicate robot tasks. For that, standardization and a common understanding of concepts in the domains involved with robots, humans and its workplaces, should be pursued. Taking this in mind, IEEE started to gather knowledge from experts in academia and industry to develop ontologies for robotics and automation, e.g., for the Core Robotics domain [8], industrial and service robots. The later with special interest to robotic surgery. As presented above, in [5] is presented the interconnections between ontologies and standards to obtain useful standardized systems to speed-up robotic development.

In [3] was presented the implementation guidelines, and first results, for a robotic orthopedic ontology, with application to a surgical procedure for hip resurfacing. Existing ontologies from the medical and the robotics fields were mapped in the OROSU ontology [2], and are considered to be the main sources of content that cover almost the knowledge needed to develop the ontology for orthopedic robotic surgery.

Figure 1, depicts the SUMO and CORA ontology parts, used in the development of OROSU. Some intra and pre-operative actions, presented in figure 2, reveal in which cases Surgical Devices and/or Algorithms can be used.

Fig. 2. Examples of pre- and intra-operative actions during orthopedic surgery.

III. Reasoning Results

Based on the actions defined in the OROSU ontology, applied to Hip Resurfacing Surgery [4], reasoning results

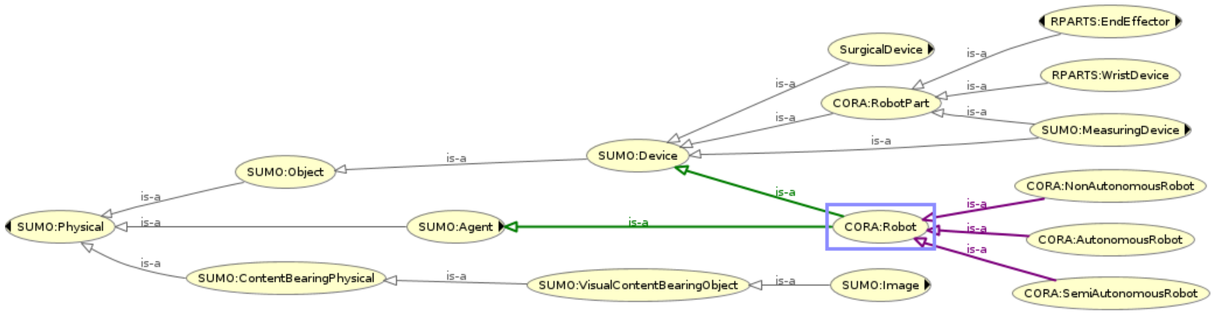

Fig. 1. The link between SUMO, CORA and OROSU ontologies.

obtained using *protégé*, are presented. For this, the reasoner HermiT was used.

Figure 4 depicts reasoning results, using the ontology, to obtain:

- the classes, in which, *SUMO:Device* are used during the reported application example;
- the instances that implement such devices, e.g., the *CORA:SemiAutonomousRobot – KUKAlwr1*.

Examples of algorithms that can be applied to the *spatialthing* class, such as *PointCloud* and/or a *Pose*, are depicted in figure 3. The later can be referenced to several frames, e.g., to the Operating Room, or CT reference frames.

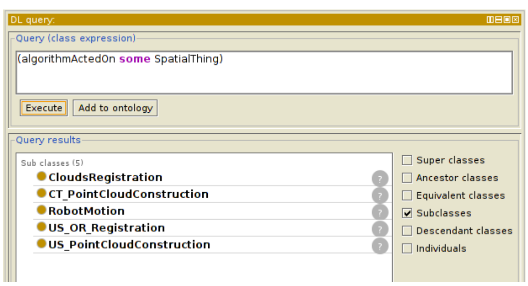

Fig. 3. Examples of algorithms that can be applied to the *spatialthing* class.

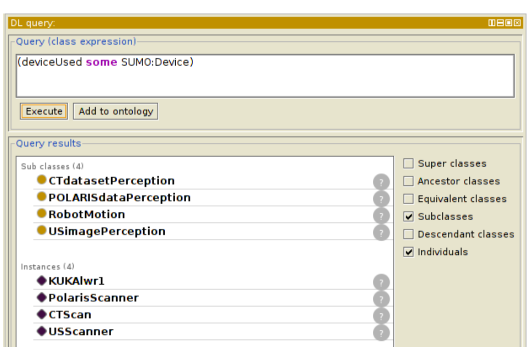

Fig. 4. Examples of *SUMO:Device* classes and instances that can be used in orthopedic surgery.

IV. Conclusions

Ontologies allow a perfect combination of surgical protocols, machine protocols, anatomical ontologies, and medical image data. With those, surgeons can perform surgical navigation with anatomical orientation, using state-of-the-art control architectures of robots, defined in the ontology.

In conclusion, the paper presented reasoning results for a domain specific ontology for robotic orthopedic surgery, i.e., Hip Surgery. Those results, enhanced the use of Imaging components during surgery, and have showed no inconsistencies in the developed framework.

Acknowledgments

This work is supported by FCT, through IDMEC, under LAETA Pest-OE/EME/LA0022.

References

[1] P. Abolmaesumi, G. Fichtinger, T.M. Peters, I. Sakuma, and G.-Z. Yang. Iintroduction to special section on surgical robotics. *IEEE Transactions on Biomedical Engineering*, 60(4):887–891, 2013.

[2] Paulo J.S. Gonçalves and Pedro M.B. Torres. Knowledge representation applied to robotic orthopedic surgery. *Robotics and Computer-Integrated Manufacturing*, (0):–, 2014.

[3] P.J.S. Gonçalves. Towards an ontology for orthopaedic surgery, application to hip resurfacing. In *Proceedings of the Hamlyn Symposium on Medical Robotics*, pages 61–62, London, UK, June 2013.

[4] P.J.S. Gonçalves, P.M.B. Torres, F. Santos, R. Antnio, N. Catarino, and J.M.M. Martins. A vision system for robotic ultrasound guided orthopaedic surgery. *Journal of Intelligent & Robotic Systems*, pages 1–13, 2014.

[5] Tamás Haidegger, Marcos Barreto, P.J.S. Gonçalves, Maki K. Habib, Veera Ragavan, Howard Li, Alberto Vaccarella, Roberta Perrone, and Edson Prestes. Applied ontologies and standards for service robots. *Robotics and Autonomous Systems*, 61(11):1215–1223, 2013.

[6] Heinrich Herre. General formal ontology (gfo): A foundational ontology for conceptual modelling. In Roberto Poli, Michael Healy, and Achilles Kameas, editors, *Theory and Applications of Ontology: Computer Applications*, pages 297–345. Springer Netherlands, 2010.

[7] Raj Mudunuri, Oliver Burgert, and Thomas Neumuth. Ontological modelling of surgical knowledge. In Stefan Fischer, Erik Maehle, and Rdiger Reischuk, editors, *GI Jahrestagung*, volume 154 of *LNI*, pages 1044–1054. GI, 2009.

[8] Edson Prestes, Joel Luis Carbonera, Sandro Rama Fiorini, Vitor A.M. Jorge, Mara Abel, Raj Madhavan, Angela Locoro, P.J.S. Gonçalves, Marcos E. Barreto, Maki Habib, Abdelghani Chibani, Sébastien Gérard, Yacine Amirat, and Craig Schlenoff. Towards a core ontology for robotics and automation. *Robotics and Autonomous Systems*, 61(11):1193–1204, 2013.

Developing and Maintaining Sub-domain Ontologies

Tamás Haidegger
* Antal Bejczy Center for Intelligent Robotics, Óbuda University, Budapest, Hungary
** Austria Center for Medical Innovation and Technology (ACMIT), Wiener Neustadt, Austria
haidegger@ieee.org

I. Generic and Domain-specific ontology development

Ontologies are considered to be a winning modelling approach in human–machine collaboration, regarding cognition and knowledge-sharing. An ontology can serve as a communication middle-layer between humans and robots. Classically, core-level and sub-level ontologies are distinguished, and this latter can be built based on certain tasks or applications [1]. Most recently, the IEEE RAS Ontologies for Robotics and Automation Working Group (ORA WG) presented a notable effort to provide a consensus-based core ontology in the domain. Their aim was to link existing ISO, IEC, etc. standards and current research efforts and new regulatory frameworks to a generic Robotics and Automation Ontology [2]. The ORA WG is comprised of over 50 members, representing over twenty countries—a cross-section of industry, academia and government. The group has spent over three years to prepare a first version (P1872/D2 Draft Standard for Ontologies for Robotics and Automation) that is now being evaluated by invited experts, organizations and the IEEE SA Standards Board. The ontologies developed by the ORA WG are intended to be merged in a broad ontology that can be further extended for cover specific applications in R&A. The goal of the work group is also to coordinate the standardization efforts with other groups to facilitate more effective information sharing, while achieving wider impact and user-base [3].

To truly support the community, numerous sub-domain ontologies are planned by the ORA WG to be linked to the core ontology. These are developed by dedicated international teams. One existing sub-group is focusing on the service robotic domain and the relationship among concepts such that humans and robots can interact to perform tasks in any environment. The complete family of the ORA will ensure a common understanding among members of the community and facilitating more efficient integration and data transfer.

II. The process of ontology development

The fundamental objective of domain-specific ontology development is to identify, develop and document the common terms and definitions within a sub-field, so that they can serve as a common reference for the R&D community. It needs to be completed at a very sophisticated way to fulfil its goals, since only high-quality ontologies can be hoped to become cornerstones of the community effort. High quality, high profile ontologies are called Exemplary Ontologies (*http://ontologydesignpatterns.org/wiki/ /Ontology:Main*). The general methodology for building ontologies specifies certain modelling principles that need to be followed in order to assure that the finished tool commits to the shared knowledge. It needs to ensure the mutual agreement among stakeholders, and increase the potential of reuse of the knowledge, allowing smooth data integration upward and downward as well. When it is targeted to develop exemplary ontologies, the following attributes need to be considered [4]:

- the ontology must be well designed for its purpose;
- shall include explicitly stated requirements;
- must meet all and for the most part, only the intended requirements;
- should not make unnecessary commitments or assumptions;
- should be easy to extend to meet additional requirements;
- it reuses prior knowledge bases as much as possible;
- there is a core set of primitives that are used to build up more complex parts;
- should be easy to understand and maintain;
- must be well documented.

An important issue towards a united robot ontology is the implementation of it, determined by its modularity, reusability and flexibility. The ORA WG thus decided to use Ontology Web Language (OWL, *http://www.w3.org/TR/owl-features*) because of its popularity in the community, and for the number of tools and reasoning engines available with it. Besides, OWL provides additional vocabulary, and facilitates greater machine interoperability of Web content. Accordingly, all sub-domain ontologies are expected to follow this convention.

Another important issue is the maintenance and curation of these ontologies, since the knowledge of each represented domain is expected to get enlarged significantly over the next couple of years. For this, several methodologies exist, typically referred to as ontology life cycle management [5, 6].

III. BUILDING A SUB-DOMAIN ONTOLOGY

The next step of the community is to set up the teams, and define more precisely the concepts of the RAS knowledge representation in the sub-domains. These groups will be developing the sub-ontologies, e.g., for the field of surgical robotics. It is essential during the particular process affecting the medical domain to:

- ensure common understanding both among members of the engineering and clinical community;
- facilitate efficient data integration from medical ontologies (e.g., OGMS – Ontology for General Medical Science, or Open Clinical);
- facilitate efficient component integration;
- facilitate more efficient information transfer among medical electrical equipment and robotic systems.

There have been some examples of medical robotic ontologies, including the REHABROBO-ONTO (Sabanci University) [7], the Surgical Workflow Ontology (SWOnt) [8], the Surgical Ontologies for Computer Assisted Surgery (SOCAS) concepts (Leipzig University) [9] or the European Robotic Surgery FP7 project (http://eurosurge.eu/). To our knowledge, there are two narrower sub-domain ontologies within the field [10]:

- Neurosurgery Robotic Ontology (NRO)
  - Lead by Politechnico di Milano;
  - Both for pre-op and intra-op phases;
  - Definition of domain's concepts (classes) by textbooks and interviews with surgeons;
  - Hierarchical organization of concepts, concept attributes, restrictions and relations among concepts (properties)
  - Definition of instances of concepts and population of the ontology.
- Total Hip Replacement Surgery Ontology
  - Coordinated by Polytechnic Institute of Castelo Branco / Technical University of Lisbon
  - Developed as part of HIPROB & ECHORD projects (www.echord.info/wikis/website/hiprob; www.echord.info/wikis/website/home).

Due to the sensitivity of the surgical domain, a more delicate ontology construction strategy is proposed, building on existing best practices [11]:

- Strategy with respect to the specialty of the application domain: taking into consideration the interdisciplinary domain requirements;
- Relying on the core ontology: identifying the interfaces and respecting the P1872 [12];
- Employing one overall strategy to identify concepts;
  - from the most concrete to the most abstract (bottom-up);
  - from the most abstract to the most concrete (top-down) or
  - from the most relevant to the most abstract and most concrete (middle-out);
- Life cycle proposal: choosing one best practice approach, e.g.:
  - ontology development process based on IEEE 1075-1995 Standard for Software Development Process or
  - IEEE 1074-1997 IEEE Standard for Developing Software Life Cycle Processes.

IV. FUTURE WORK

It is believed that the proper approach towards ontology engineering will lead to a set of mid-level and sub-domain ontologies truly useful and applicable to the current cognitive robotics research, particularly in the domain of service robots and medical robots within. The ORA WG is about to set up a dedicated WG to investigate and develop the relevant sub-domain ontology for surgical robot and computer-integrated surgical systems with the active support of the global research community.

ACKNOWLEDGMENT

T. Haidegger is a Bolyai Fellow of the Hungarian Academy of Sciences.

REFERENCES

[1] N. Guarino, "Semantic matching: Formal ontological distinctions for information organization, extraction, and integration". Information Extraction A Multidisciplinary Approach to an Emerging Information Technology. Springer, Berlin/Heidelberg, pp. 139–170, 1997.

[2] S. Lemaignan et al. "Bridges between the Methodological and Practical Work of the Robotics and Cognitive Systems Communities—From Sensors to Concepts, chapter Towards Grounding Human–Robot Interaction". Intelligent Systems Reference Library. 2012.

[3] Fiorini, S. R., Carbonera, J. L., Gonçalves, P., Jorge, V. A., Rey, V. F., Haidegger, T., ... & Prestes, E. (2014). Extensions to the core ontology for robotics and automation. Robotics and Computer-Integrated Manufacturing. (http://www.sciencedirect.com/science/article/pii/S0736584514000659)

[4] Ontology Design Patterns (ODP), NeOn project, http://ontologydesignpatterns.org, 2014.

[5] Steffen Staab: "Ontology Lifecycle". ISWeb Lecture, Universität Koblenz, Landau, 2007.

[6] N. Noy, T. Tudorache, C. Nyulas, and M. Musen, 'The ontology life cycle: Integrated tools for editing, publishing, peer review, and evolution of ontologies'. In Proc. of the American Medical Informatics Association Annual Symp. (AMIA), pp. 552, 2010.

[7] Z. Dogmus, A. Papantoniou, M. Kilinc, S. A. Yildirim, E. Erdem, and V. Patoglu. "Rehabilitation robotics ontology on the cloud." In Proc. of the IEEE Intl. Conf. on Rehabilitation Robotics (ICORR), pp. 1–6, 2013.

[8] R. Mudunuri, O. Burgert and T. Neumuth, "Ontological Modelling of Surgical Knowledge." GI Jahrestagung, vol. 154, pp. 1044–1054, 2010.

[9] P. J. S. Gonçalves and P. M. B. Torres, "A Survey on Biomedical Knowledge Representation for Robotic Orthopaedic Surgery." In Robot Intelligence Technology and Applications, vol. 2, pp. 259–268. Springer International Publishing, 2014.

[10] T. Haidegger, M. Barreto, P.J. S. Goncalves, M. K. Habib, S. V. Ragavan, H. Li, ... and E. Prestes, "Robot ontologies for sensor-and Image-guided surgery". In Proc. of the IEEE Intl Symposium on Robotic and Sensors Environments (ROSE), pp. 19–24, 2013.

[11] "OntoWeb deliverable, 1.4." Universidad Politecnia de Madrid, 2002.

[12] IEEE RAS Joint Workgroup for Ontologies for Robotics and Automation, "P1872/D2 Draft Standard for Ontologies for Robotics and Automation," 2014.

Ontological Query Answering about Rehabilitation Robotics

Zeynep Dogmus Volkan Patoglu Esra Erdem

***Abstract*— We introduce a novel method to answer natural language queries about rehabilitation robotics, over the formal ontology REHABROBO-ONTO. As part of our method, 1) we design and develop a novel controlled natural language for rehabilitation robotics, called REHABROBO-CNL; 2) we introduce translations of queries in REHABROBO-CNL into SPARQL queries, utilizing a novel concept of query description trees; 3) we use an automated reasoner to find an answer to the SPARQL query. To facilitate the use of our method by experts, we develop an intelligent, interactive query answering system, and make it available on the cloud.**

I. INTRODUCTION

The first formal ontology about rehabilitation robotics, called REHABROBO-ONTO, has been recently designed and developed in OWL (Web Ontology Language) [1], [2], and made available on the cloud [3], [4], with the goal of facilitating access to various kinds of information about the existing rehabilitation robots. Indeed, such a formal ontology allows rehabilitation robotics researchers to learn various properties of the existing robots and access to the related publications to further improve the state-of-the-art. Physical medicine experts also can find information about rehabilitation robots and related publications to better identify the right robot for a particular therapy or patient population. Such requested information can be obtained from REHABROBO-ONTO by expressing the requested information as queries.

With this motivation, we introduce novel methods for representing and answering queries about rehabilitation robots over REHABROBO-ONTO in such a way that the queries are expressed in natural language and their answers are obtained using the state-of-the-art automated reasoners over REHABROBO-ONTO. To facilitate the use of our methods by experts, we develop an intelligent, interactive query answering system, and make it available on the cloud via Amazon web services.

A longer version of this paper will appear in Proceedings of the Sixth International Conference on Knowledge Engineering and Ontology Development [5].

II. METHOD

Our method for answering natural language queries about rehabilitation robots consists of the following four stages.

This work is partially supported by Sabanci University IRP Grant and TUBITAK Grant 111M186.

Z. Dogmus, V. Patoglu, E. Erdem are with Faculty of Engineering and Natural Sciences, Sabancı University, İstanbul, Turkey {zeynepdogmus,esraerdem,vpatoglu}@sabanciuniv.edu

A. Expressing queries in natural language

To overcome the ambiguities in the vocabulary and grammar of natural languages, we design and develop a novel controlled natural language (CNL), called REHABROBO-CNL, for representing queries about rehabilitation robots. By this way, the users are not required to be familiar with the underlying formal query language to be used with automated reasoners. Here is a sample query in REHABROBO-CNL:

> What are the robots that target some shoulder movements with actuation='electrical' and (with transmission='cable drive' or with transmission='direct drive')?

A CNL is a subset of a natural language with a restricted vocabulary and grammar. A part of the grammar of REHABROBO-CNL is shown in Table I. The italic functions in the grammar are used to extract relevant information from REHABROBO-ONTO. These ontology functions are described in Table II.

A CNL can be viewed as a formal language and thus can be converted into a logic-based formalism required for automated reasoning. We introduce methods for parsing and translating natural language queries in REHABROBO-CNL into formal SPARQL queries [6].

B. *Parsing* REHABROBO-CNL *queries as trees*

Parsing a REHABROBO-CNL query utilizes a novel tree structure, called a Query Description Tree (QDT). A QDT is a rooted, directed tree that consists of five types of nodes:

- root-node: Represents the sort of the query.
- that-node: Represents a relative clause beginning with "that".
- with-node: Represents a relative clause beginning with "with".
- and-node: Represents a conjunction.
- or-node: Represents a disjunction.

Every root/that/with-node characterizes a phrase and a type/instance. An and/or-node cannot be a leaf. For each path from the root node to a leaf node, there can be at most one and-node and one or-node. With-nodes are leaves only. That-node has one child only.

For instance, consider the REHABROBO-CNL query presented in the previous section. The QDT for this query is presented in Figure 1.

C. *Translating QDTs into* SPARQL *queries*

The QDT representing the query, in fact, represents a concept. While creating a query, we define a new concept

TABLE I: The Grammar of REHABROBO-CNL

QUERY →	WHATQUERY QUESTIONMARK
WHATQUERY →	What are the $Type()$ GENERALRELATION
GENERALRELATION →	SIMPLERELATION NESTEDRELATION*
SIMPLERELATION →	(that RELATIVECLAUSE)+
SIMPLERELATION →	WITHRELATION
NESTEDRELATION →	(and ((LP SIMPLEDISJUNCTION RP) — SIMPLECONJUNCTION))*
SIMPLEDISJUNCTION →	(SIMPLERELATION or)* SIMPLERELATION
SIMPLECONJUNCTION →	(SIMPLERELATION and)* SIMPLERELATION
RELATIVECLAUSE →	$Verb()$ (some | all | the) $Type()$
WITHRELATION →	with $Noun()$ EQCHECK $Value()$+
QUESTIONMARK →	?

TABLE II: The Ontology Functions

$Type()$	Returns the types that correspond to concept names. They are: Robots, movements, users, publications and metrics.
$Verb()$	Returns the verbs that correspond to object properties between concepts. Returns both active and passive forms of these verbs. Active forms of these verbs are: Target, evaluate, reference, own.

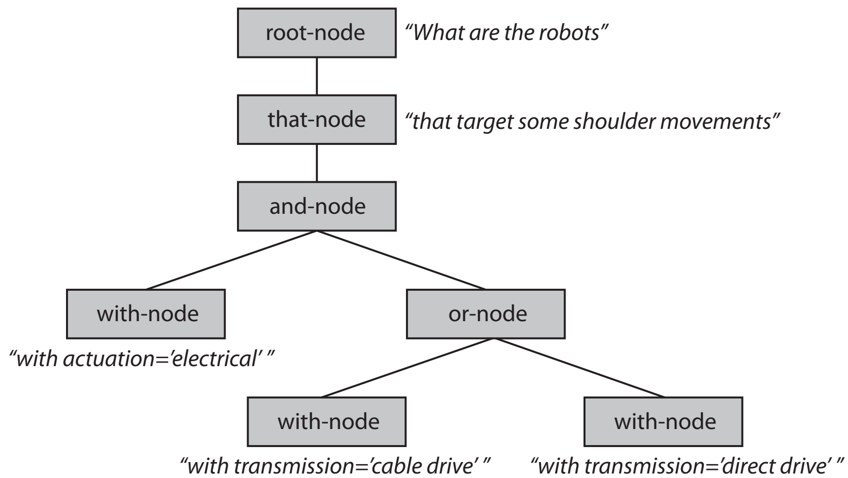

Fig. 1: Tree representation of the sample query.

and search for its instances. Retrieved instances that fit our description are the answers to our query. Therefore, by a depth-first traversal of a QDT (Algorithm 1), we can represent the corresponding concept formally. For instance, the QDT in Figure 1 characterizes the following Description Logics (DL) concept:

$$\mathtt{Robot} \sqcap \exists\mathtt{targets.ShoulderMovements} \sqcap \exists\mathtt{actuation.}\{\mathtt{electrical}\} \sqcap (\exists\mathtt{transmission.}\{\mathtt{cabledrive}\} \sqcup \exists\mathtt{transmission.}\{\mathtt{directdrive}\}).$$

To obtain a SPARQL concept from a DL concept, we utilize some of the existing translations in related publications, such as [8] and [7]. We also introduce some novel transformations, for inverse roles, complement and universal restrictions. Some transformation examples are shown in Table III. The transformations without a citation are the novel transformations. By these transformations, for instance, the DL concept above can be transformed into the following SPARQL concept:

```
?robot1 rdf:type rr:RehabRobots.
?robot1 rr:targets ?movement1.
?movement1 rdf:type rr:ShoulderMovements.
?movement1 rr:has_Actuation 'electrical'.
{?movement1 rr:has_Transmission 'cable drive'.}
UNION
{?movement1 rr:has_Transmission 'direct drive'.}
```

After we transform a DL concept into a SPARQL concept, we can construct a SPARQL query as follows. We start with a PREFIX part and we declare the namespace (the location of an ontology on the Web) of REHABROBO-ONTO. Next, we continue with a SELECT clause, where we specify to display the names of the instances to the users. After that, we include the SPARQL concept:

```
PREFIX rdf: <http://.../22-rdf-syntax-ns#>
PREFIX rr: <http://.../RehabOnto.owl#>

SELECT DISTINCT ?name WHERE {
  ?robot1 rr:has_Name ?name.
  <SPARQL concept>
}
```

TABLE III: DL to SPARQL Transformation Examples

Constructor	**DL**	**SPARQL**
Concept [7]	`Robot`	`?x rdf:type ns:RehabRobots.`
Role [8]	`targets`	`?x ns:targets ?y.`

Complement	$\neg\exists$`name.{AssistOn}`	`FILTER NOT EXISTS {`
		`?x ns:has_Name 'AssistOn'.`
		`}`

Inverse Role	$\exists$`targets`$^{-}$`.Robot`	`?x ns:targets ?y.` `?x rdf:type ns:RehabRobots.`
Existential Restriction [8]	$\exists$`targets.ShoulderMovements`	`?x ns:targets ?y.` `?y rdf:type ns:ShoulderMovements.`

Universal Restriction	$\forall$`reference.Robot`	`?x rr:reference ?y.`
		`?y rdf:type rr:RehabRobots.`
		`FILTER NOT EXISTS {`
		`FILTER NOT EXISTS {`
		`?x rr:reference ?y2.}`
		`?y2 rdf:type rr:RehabRobots.}`

D. *Answering* SPARQL *queries using* PELLET

Once we obtain a SPARQL query, we can use the description logics reasoner PELLET [9] to find answers to queries, through the Jena framework.

E. *An interactive, intelligent user-interface*

We design and develop an interactive, intelligent user-interface to guide users to express their natural language queries about rehabilitation robots in REHABROBO-CNL, and to present the answers to their queries with links to detailed information.

The main user interface for querying includes a drop-down list, showing the possible ways to begin a query. Then, according to the user's choices, it provides different types of features. It provides auto-completion to help users enter values for nouns that correspond to data properties of type string. If the user should choose a concept among a hierarchy, then it displays an accordion view and enables the user to click on the option s/he wants. In addition, it allows multiple selection of values for relational properties. For functional properties, user is able to select multiple items for inequality. User can choose a number of options among "less than or equal", "more than or equal", "equal" and "not equal" while entering values for a data property of type integer or float. Figure 2 illustrates some parts of constructing the query "What are the robots that target some wrist movements with actuation='series elastic'?" with this interface.

The system is available on the cloud via Amazon web services.[1]

[1]`http://ec2-54-228-52-230.eu-west-1.compute.amazonaws.com/rehabrobo/`

III. RELATED WORK

There are ontology systems, like QACID [10], PowerAqua [11], FREyA [12], with natural language interfaces. These systems also take natural language queries, translate a query into a SPARQL query, and use a query engine to find an answer over specified ontologies. However, these systems are restricted to simple forms of queries (e.g., that do not involve negation, disjunction or relative clauses). Our query language and query answering methods allow more complex forms of queries.

To eliminate the ambiguity of natural language queries and to allow a larger variety of queries [13], [14] consider CNLs. Our work is similar to these related work since we also consider queries in a CNL, but we target a different domain and more general forms of queries (e.g., that involve negations, quantifiers, or nested clauses).

IV. CONCLUSION

We have introduced a novel method to answer natural language queries about rehabilitation robotics, over the first formal rehabilitation robotics ontology REHABROBO-ONTO. We have developed an intelligent, interactive query answering system, using Semantic Web technologies, and deployed it on the cloud via Amazon web services. Our ongoing work involves evaluation of this system by both rehabilitation engineers and physical medicine experts.

REFERENCES

[1] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen, "From shiq and rdf to owl: the making of a web ontology language," *J. Web Sem.*, vol. 1, no. 1, pp. 7–26, 2003.

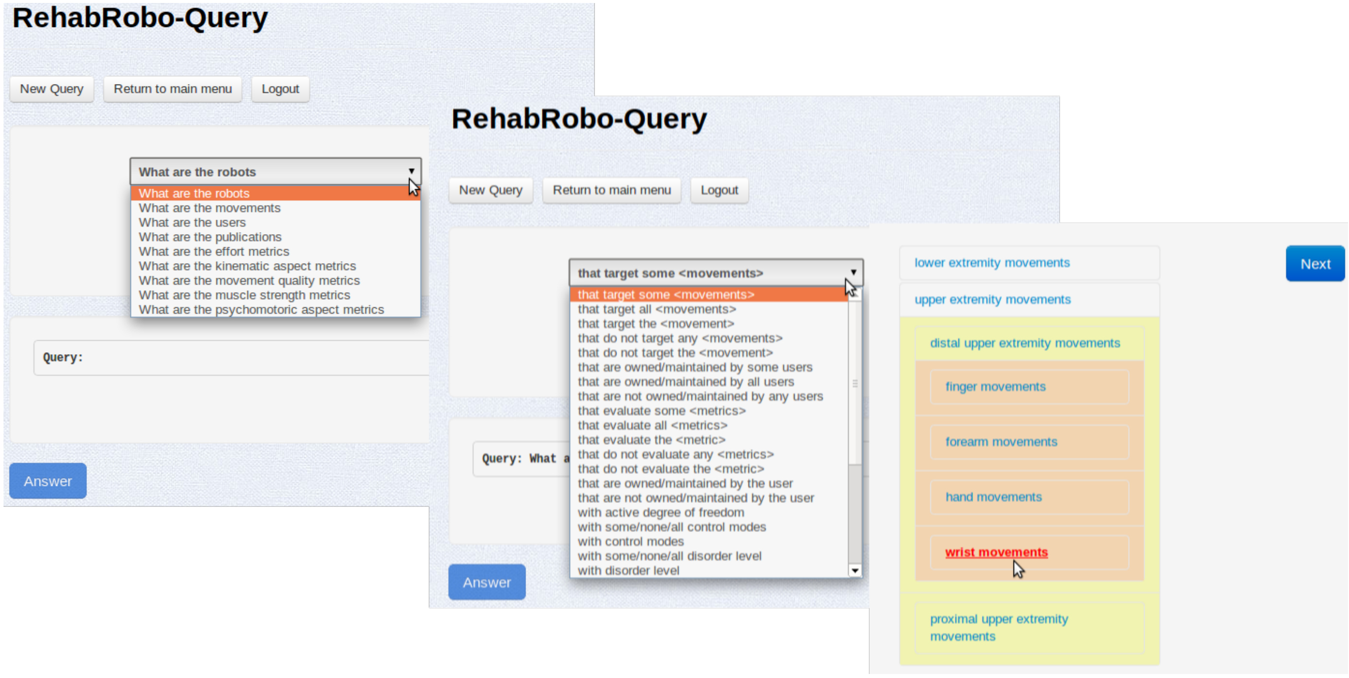

Fig. 2: Constructing a query with the guide of an interactive, intelligent user interface.

[2] G. Antoniou and F. van Harmelen, "Web ontology language: Owl," in *Handbook on Ontologies*, 2004, pp. 67–92.

[3] Z. Dogmus, A. Papantoniou, M. Kilinc, S. A. Yildirim, E. Erdem, and V. Patoglu, "Rehabilitation robotics ontology on the cloud," in *Proc. of the 13th International Conference on Rehabilitation Robotics (ICORR)*, 2013.

[4] Z. Dogmus, G. Gezici, V. Patoglu, and E. Erdem, "Developing and maintaining an ontology for rehabilitation robotics," in *Proc. of KEOD*, 2012, pp. 389–395.

[5] Z. D. V. Patoglu and E. Erdem, "Answering natural language queries about rehabilitation robotics ontology on the cloud," in *Proc. of KEOD*, 2014.

[6] E. PrudHommeaux, A. Seaborne, *et al.*, "Sparql query language for rdf," *W3C recommendation*, vol. 15, 2008.

[7] D. Y. S. Fernandes, "Using semantics to enhance query reformulation in dynamic distributed environments," Ph.D. dissertation, Federal University of Pernambuco, 2009.

[8] G. Orsi, "Context based querying of dynamic and heterogeneous information sources," Ph.D. dissertation, Politecnico di Milano, 2011.

[9] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 51 – 53, 2007.

[10] O. Ferrández, R. Izquierdo, S. Ferrández, and J. L. Vicedo, "Addressing ontology-based question answering with collections of user queries," *Information Processing and Management*, vol. 45, no. 2, pp. 175 – 188, 2009.

[11] V. Lopez, M. Fernández, E. Motta, and N. Stieler, "PowerAqua: Supporting users in querying and exploring the semantic web," *Semantic Web*, vol. 3, no. 3, pp. 249–265, 2012.

[12] D. Damljanovic, M. Agatonovic, and H. Cunningham, "FREyA: an interactive way of querying linked data using natural language," in *Proceedings of the 8th international conference on The Semantic Web*, ser. Proc. ESWC, 2012, pp. 125–138.

[13] E. Erdem, Y. Erdem, H. Erdogan, and U. Oztok, "Finding answers and generating explanations for complex biomedical queries," in *Proc. AAAI*, 2011.

[14] R. Valencia-García, F. García-Sánchez, D. Castellanos-Nieves, and J. Fernández-Breis, "OWLPath: An OWL ontology-guided query editor," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 41, no. 1, pp. 121–136, 2011.

Algorithm 1: transform

Input : A tree T representing the concept that the user described

Output: A DL concept description Q that represents the concept in T

// $n.class$ denote associated class of a node n
// $n.children$ denote children of a node n
$Q \leftarrow \emptyset$;
$n \leftarrow$ first (root) node in T;
if n *is a root-node* **then**
  $Q \leftarrow Q \sqcap n.class$;
  foreach *child node* $c \in n.children$ **do**
    $Q \leftarrow Q \sqcap transform(c)$;
else if n *is a that-node* **then**
  $Q \leftarrow Q \sqcap transformThatNode(n)$;
else if n *is a with-node* **then**
  $Q \leftarrow Q \sqcap transformWithNode(n)$;
else if n *is an and-node OR* n *is an or-node* **then**
  $tempQ \leftarrow \emptyset$;
  foreach *child node* $c \in n.children$ **do**
    if n *is an and-node* **then**
      $tempQ \leftarrow tempQ \sqcap transform(c)$;
    else
      $tempQ \leftarrow tempQ \sqcup transform(c)$;
  $Q \leftarrow Q \sqcap (tempQ)$;
return Q

Towards a Standard for Grounding Symbols for Robotic Task Ontologies

Chris Paxton, Jonathan Bohren, Gregory D. Hager

***Abstract*— We discuss the "symbol grounding problem" in the development of robotic tasks and how different types of grounding can affect the generality of automatic planning and recognition algorithms. Grounding is the mechanism through which task symbols get associated with concrete meanings and effects on the world. We discuss the implications of different strategies when grounding symbols in robotic task models. In practice, the extent to which a symbol is grounded lies on a continuum between "implict" and "explicit" grounding. *Implicitly grounded* symbols are those which have requirements and effects that are not represented in the symbol system. This can occur when the effects of a symbol are only determined by the result of executing an opaque action or because the task model is constructed from a set of simple rules that lack the expressiveness to wholly capture the task requirements. Alternatively, symbols can be *explicitly grounded* by describing their requirements and effects on the world in terms of real physical quantities. Such grounding enables a system to better reason about physical entities in the world, their properties, and their behavior. This approach comes with its own limitations, due to increased complexity and difficulty creating accurate models. We discuss a handful of different grounding approaches in terms of a case study with a simulated multi-arm manipulation task. These considerations will help develop standards of grounding for task ontologies and describe how robust they make a given task model.**

I. Introduction

The artificial intelligence and task planning communities have made great strides in developing formalisms and frameworks for symbolically modeling complex concepts. However, building grounded symbolic knowledge for such systems in the context of unstructured robotic applications is still a challenging and often poorly-defined task. A large part of this challenge is finding a balance between the ease with which a task can be specified and the generality of the model of that task in practice. The way in which symbols are grounded becomes even more significant when trying to learn such symbolic models from demonstration since it necessitates systems which can provide the needed contextual and relational information.

We discuss a handful of different grounding approaches in terms of a case study with a simulated multi-arm manipulation task. In this context, *symbol grounding* is the mechanism through which task symbols get associated with concrete meanings and effects on the world. In practice, the extent to which a symbol is grounded lies on a continuum between "implicit" and "explicit" grounding. This continuum is based on ideas related to the "symbol grounding problem" in cognitive science as described by Harnad [1]. In that context, the symbol grounding problem" characterizes the challenges

C. Paxton, J. Bohren, and G. D. Hager are from the Department of Computer Science, Johns Hopkins University, 3400 N. Charles St. Baltimore, MD 21218-2686, USA (email: {cpaxton3@jhu.edu, jbo@jhu.edu, hager@cs.jhu.edu}).

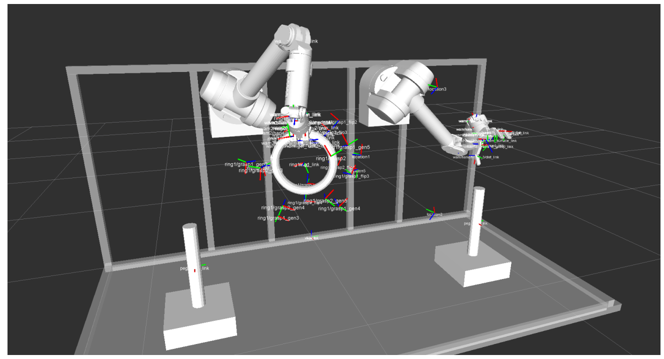

Fig. 1: Two simulated Barrett WAM arms after handing off a ring between two pegs in our example task. Image shows coordinate frames for objects, waypoints, and grasp points.

in representing the underlying meaning of cognitive symbols. Harnad also emphasizes that grounding of a symbol is a *systematic* property, and that "an isolated ('modular') chunk cannot be symbolic" [1]. As such, the amount and specificity to which a symbol is described and related to other symbols represents another dimension in which grounding can vary.

We define *implicitly grounded* symbols as those which have requirements and effects which only poorly approximate the "true" meaning of the symbols. In this case, the "true" meaning is that which is intended by a human designer. Symbols can be implicitly grounded when the effects of a symbol are only determined by the result of executing an opaque action or because the task model is constructed from a set of simple rules that lack the expressiveness to wholly capture the task requirements.

Alternatively, symbols can be *explicitly grounded* by describing their requirements and effects on the world in terms of real physical quantities. Such grounding enables a system to better reason about physical entities in the world, their properties, and their behavior. This approach comes with its own limitations, due to increased complexity and difficulty creating accurate models.

As an example, consider modeling how a robot should place an object on a table. Before an object can be placed on a table, we understand that there must be space for it: the table must be "clear-enough" for the object to be placed. In this case, the grounding problem lies in how different notions of "clearness" are connected to real-world measurable attributes and effects.

One strategy for grounding the concept of "clear-enough" is to implement some opaque function `TableIsClearDetector` which simply emits "yes" or "no" as a result. In isolation, this "clear-enough" symbol has no meaning besides its functional signature. As such, the meaning of the "clear-enough" symbol is *implicitly* *grounded* by the details of this computation. Without the ability to reason about the underlying implementation, however, it is impossible to introspect on either the details about why it emitted a certain result or on the consequences of this result.

Another approach for defining "clear-enough" is to define "clearness" as a set of one or more simpler computable symbols, and then describe the real-world attributes and effects of these simpler symbols. For example, we could say that a table is "clear-enough" if no objects intersect with a volume up to 0.1 meters above the given table. As long as these symbols completely describe the underlying task, we can accurately reason about what makes a table "clear-enough" and the task is *explicitly grounded*.

However, consider a situation where a relatively small object like a vase might be occupying some space at the center of the table. This vase prevents the table from being considered "clear-enough", even though there might still be space to place the given object. This is still an implicit grounding of the task, because it approximates and does not completely describe task constraints.

In a true explicit grounding of the task, we might use a physics simulation to simulate placing an object on the table in different positions. If the object sits comfortably on the table, then the table was "clear-enough" to complete our task. In our case, this is exactly describing the task requirements as they would be interpreted by a human.

In the subsequent sections, we describe different strategies for grounding parts of a symbolic task model, where each strategy entails a different method for providing meaning to symbols that an automated system can reason about. Our goal in this paper is to describe the necessary attributes of a grounding approach for task descriptions that will allow generalizable task performance.

As a concrete example task, we consider two 7DOF Barrett WAM arms which need to cooperate to pick up a ring and transfer it from one peg to another in a simulated environment. As shown in Figure 1, neither arm can reach both pegs, so the high-level controller needs to coordinate a hand-off between the manipulators. The task was modelled after a similar peg-transfer task used in robotic surgical training [2] which requires bimanual coordination.This case study is described further in Section IV.

II. Background

Recent work has explored a variety of powerful task planners and systems that allow robots to perform complex tasks under assumption that if given a "sufficiently sophisticated" model, robots will be able to solve real-world problems. Often, previous work has used Hierarchical Task Networks (HTNs) or similar formalisms as a way of describing each high-level task as a series of sub-tasks. Some previous work has looked at using HTNs together with lower-level control [3], [4]. Another system, KnowRob, uses task ontologies to enable robot planning for complex tasks which can generalize to different robotic platforms [5]. Other work has built robust task planners that have proven an ability to generate complex plans given more abstract constraints, without an explicit, step-by-step task plan [6]. These systems invariably rely on a large number of carefully designed high level predicates to function, generally making them examples of top-down implicit grounding.

Work in the learning from demonstration community has often produced models that respond to low-level primitive features of the environment to mimic an expert's behavior [7]. recent work in Gaussian process inverse reinforcement learning could mimic an expert's response to a highway driving task or to a robot arm navigation task, for example [8]. These approaches mimic feature responses without any knowledge of effects or long term goals, and are as such examples of bottom-up implicit grounding.

Such symbolic representations are only as useful as the level to which their implications on the state of the world is modeled. When the expressiveness of these models match the "true" model of the task, it enables the planning system to synthesize solutions to problems which might not have even been predicted by the system's developers [6]. The challenge of implementing symbolic reasoning systems in practice has led to an interest in symbol grounding through learning [9]. The necessary level to which symbols must be grounded in real data and constraints is unclear and as a result roboticists often develop models which are only sufficient for the execution of pre-defined tasks with domain-specific parameters.

III. Definitions

We discuss three main strategies for grounding a task in a set of symbols. *Symbols* are arbitrary labels which represent information, expressed as logical *predicates*.Predicates accept a fixed number of parameters that must belong to certain classes in order to be valid. Specifics of the different grounding strategies are discussed below.

A. Top-Down Grounding

Top-down grounding means that most of the knowledge necessary for a given sub-task is provided either by specialized sub-routines designed for performance of that sub-task, or is specified beforehand by the user. The process providing this knowledge is opaque to the task model, which makes assumptions based on whether specified sub-tasks were successful. This strategy for modeling hierarchically-structured tasks yields models which obscure the fundamental constraints which necessitated the task structure in the first place. A top-down description of a task begins with a root node capturing the complete task, which can be successively broken down into smaller, re-useable symbolic units until decomposed into sub-tasks that are known to be performable by the lower-level robotic system. These sub-tasks are treated as a black box. While this is an intuitive approach for building a hierarchical task plan, when it is used to build a task model it tends to capture little more expressiveness than a hierarchical concurrent finite state machine [10]. This is because the grounding of the high- and mid-level concepts is only modeled implicitly in the capabilities of the robot.

This is the simplest strategy for reasoning about the world, and is often used in practice because it allows human engineering and careful system design to overcome limitations in perception and reasoning. Note that this approach makes sense for many robotic applications: legal grasp points, for example, are often specified beforehand because computing force-closure grasps on the fly for every object the robot may need is infeasible.

B. Bottom-Up Grounding

We say that bottom-up symbolic grounding is when a task plan is constructed as a set of simple, reusable primitive statements that build up to higher-level goals. In a bottom-up description of a task, Bottom-up grounding is commonly used in the AI and learning-from-demonstration communities.

Using our case study as an example, a specification for placing a ring on a peg would involve some combination of spatial relations and proximity symbols constraining the position of the robot end effector or the ring relative to a peg. While such predicates are easy for a human familiar with the given task to reason about, these do not actually model the reason why a given amount of "nearness" is important.

It is very difficult to build effective bottom-up systems. Dantam et al. used a context-free grammar to describe complex human-machine interactions which simultaneously reasons about symbolic states and continuous-space controllers [11]. This system uses an intricate task model that responds to low-level sensory and motor tokens indicating movement and collisions with obstacles. A successful example of a bottom-up plan is previous work by McGann et al.: it is able to generalize and achieve goals without an explicit, step-by-step task plan thanks to a simple internal model of the task that captures a broad range of possibilities [6].

It may make the most sense to use machine learning to decide which low level predicates are the most important since there are so many of them. This is generally the approach taken by work in inverse reinforcement learning and in imitation learning, which learns a policy approximating an expert demonstration based on a number of low-level features.

C. Explicit Grounding

We discuss explicit grounding as an alternative to the above. Instead of simply defining predicates in abstract terms with opaque representation or arbitrary parameterization as in the previous sections, we can ground them based on the constraints of the task. Instead, we ensure that the specific conditions necessary for an action are modeled by a given symbol.

One approach utilized in existing systems is the use of a *reachability* predicate instead of one modeling *nearness*. This explicitly captures the notion that an object's proximity is not the important feature, but rather its location in a given configuration space. Such predicates are computable based on the physics of the world and the robots' kinematics.

The same philosophy can be applied to the ring-peg placement sub-task. Instead of representing the requirements for placing a ring on a peg as a set of instance-specific spatial proximities and orientations, it can be based on more fundamental requirements. In this case, drop locations should be those such that the ring is captured by the peg, which isgenerated via a low-fidelity physical simulation.

The challenge in grounding such physically-based symbols lies is reducing their representation to a feature space which can be used for reasoning in a symbolic planner or task recognition framework. Regardless of this challenge, such constraint-centric representations may be key to building task ontologies which generalize not just to similarly capable robots, but also to similar tasks.

A slightly less-challenging aspect to demonstrate, however, is an explicitly-grounded symbolic representation of the goal critereon which can be used to verify the success of a ring placement.

IV. Case Study

The simulation was implemented using ROS [12] and the Gazebo Simulator. Since our interest is in the grounding of symbols, we used a pre-defined task sequence implemented as a finite state machine. Individual states each define a single step in a procedure. Our simulated environment is modeled in real-time by sets of boolean-valued propositional statements (predicates). Each predicate is composed of a label token and a tuple of relevant arguments.

Low level motion planning is completed by MoveIt. We implemented actions such as `grab[object, arm]` action as a move parameterized by an object, and given a list of candidate locations. The system selects the candidate location that will bring the arm close to the object and then closes the gripper. The `move[arm, location]` action uses MoveIt to plan a path to a specific location.

Actions were implemented as generic SMACH nodes parametrized by objects and entities in the world [13]. Previous work has constructed SMACH machines with an HTN planner [14], but this was deemed unnecessary for our case study as it is intended for illustrative purposes.

A. Task Plan

While we envision the task as a series of HTN-style low-level action primitives defined by pre- and post-conditions, we explicitly create the task model for three different experiments describing different levels of symbolic grounding rather than using an HTN planner. It is possible to generate state-machine-based task plans from an HTN as demonstrated previous work [14].

We create separate plans exemplifying different strategies for grounding the following steps in information available to the simulated robots:

1) Arm 1 reaches towards the ring.
2) Arm 1 grabs the ring.
3) Arm 1 lifts the ring off of the first peg.
4) Arm 1 moves the ring to a location where it can be grabbed by Arm 2.

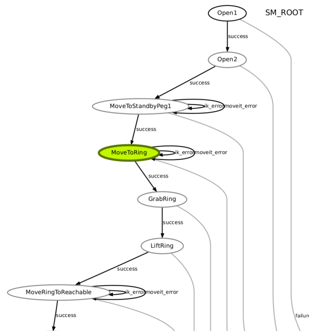

Fig. 2: Section of the state machine describing the top-down grounding of the peg transfer task.

5) Arm 2 reaches towards the ring.
6) Arm 2 grabs the ring.
7) Arm 1 releases the ring.
8) Arm 1 moves out of the way, so that it does not interfere with Arm 2.
9) Arm 2 moves to a position so that it can drop the ring on the peg.
10) Arm 2 releases the ring.

This plan was implemented as a SMACH state machine for each version of the task. The state machine changes slightly for each of the three strategies we used to ground the procedure, but remains largely the same. Each node in the state machine is parameterized by object, waypoint, and predicate goals.

Several decisions informed the design of this task model: in particular, the task model needed to take into account the limits of our low-level controllers when planning specific movements. In particular, movements that involve physical collisions with the world often require a different motion planner from that used for general obstacle avoidance behavior. This means that we have separate actions for "MoveToStandbyPeg1", when the first WAM arm moves to a position over the ring to pick it up, and "MoveToRing", when the same WAM arm moves to a grasp point on the ring.

B. Symbolic Implementation

We designed a system for producing and aggregating a variety of high- and low-level symbols representing knowledge of the world. Different modules in this system produce predicates with various levels of grounding, informed by the symbol grounding strategies described in Section III.

We define many of the predicates used in this case study in abstract terms such as geometric relationships or specific joint positions; however, we ground some predicates using world knowledge to ensure that the specific conditions necessary for an action to complete have been met. There are two types of grounded predicates in our system: those that are grounded in terms of their physical effects and those that are grounded in terms of the internal representation of the world for the robot.Grounded predicates for physical properties are predicates such as `drop_point` and `reachable`, which are computable based on the physics of the world and the robots' kinematics. Grounded Predicates for system capabilities gain their meaning from processing capabilities of the system. Perceptual and motion planning limitations may not physically limit the robot, but they effect the possible success of different actions and need to be taken account in task-level planning.

C. Detailed Single Symbol Example

The representational distinctions of the three different approaches which we describe can be illustrated by considering the `ring-on-peg` symbol. This symbol corresponds to the succesful placement of a given ring on a peg.

Each of the following examples are written as declarative descriptions which result in the final goal state of a given ring being securely placed on a peg. Note that all three examples are written in the context of the same framework. This framework contains four types of symbols: *actions*, *objects*, *predicates*, and *grounded predicates*. Objects are show below in light weight, and they can hold references to other objects. Actions and predicates take arguments and are written with an in-line notation to improve readability.

1) Top-Down: The first, and simplest model which can complete the task and indicate that a ring has been placed on a peg is a top-down model which *only* grounds the meanings of the post-conditions of an action implicitly through the implementation of the action.

Ths following declarations are suitable for defining a simple 3-state automaton which results in the predicate `ring ON peg` being true if all actions are completed.

```
ACTION: GRASP ring WITH arm
PRE: arm IDLE
POST: peg GRASPED BY arm

ACTION: MOVE ring OVER peg WITH arm
PRE: peg GRASPED BY arm
POST: ring OVER peg

ACTION: RELEASE ring ONTO peg WITH arm
PRE: ring OVER peg
POST: ring ON peg
      arm IDLE
```

While simple to implement, this model completely obscures the meaning of the predicate post-conditions. There is no explicit connection to the physical world.

2) Bottom-Up: We can instead write out a simple bottom-up predicate which models when a ring is considered properly on a peg.

The following is simply an expansion of the `ring ON peg` predicate which grounds the notions in physical quantities. These physical quantities could be used to process sensor data to implement this model or used as part of a combined task and motion planner to determine how to position the ring so that it is on a given peg.

```
PREDICATE: ring ON peg
REQ: ring BELOW peg.top
     ring ABOVE peg.bottom
     ring WITHIN 0.1 OF peg IN x y
```

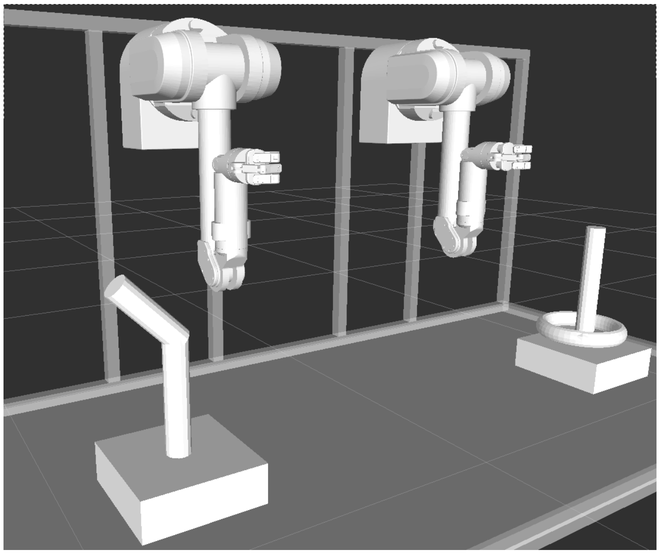

Fig. 3: Alternate version of the peg transfer task with a differently shaped peg.

Unfortunately, it is easy to imagine counter-examples which would invalidate the correctness of this predicate. Any non-straight peg or asymmetric ring would cause this predicate to evalute to false positives.

3) Effect-Based: Finally, we can define a set of predicates which attempt to capture the fundamental physical meaning of a ring being on a peg. Namely, the notion of physical intersection and the notion of constraint under perturbations normal to the peg profile.

```
PREDICATE: region_a INTERSECTS region_b
REQ: region_a INTERSECT region_b NOT EMPTY

PREDICATE: obj_a CONSTRAINED BY path IN region
REQ: sub_path := path CONTAINED IN region
     planes := NORMAL PLANES ALONG sub_path
     ring.pose BOUNDED IN planes

PREDICATE: ring ON peg
REQ: peg.mesh INTERSECTS ring.convex
     ring CONSTRAINED BY peg.profile IN ring.convex
```

While this model would be dramatically more robust, it would also be more complex to design and implement.

D. Experiments

We describe task models that work on a straightforward peg task, and then apply these models to a new version of the task with pegs of a slightly different shape to show how well different models generalize and how well different predicates apply to the new task context. The top of this alternate peg is bent at a 45 degree angle, as per Figure 3. All of the grounding strategies we described are sufficient to perform the peg-transfer task under normal circumstances.

We find top-down grounding to be a straightforward way to produce a program that will reliably complete the described ring-transfer task. In fact, top-down grounding is reasonably robust to errors: when the Barrett arms collide with an environment object, the underlying motion controllers will report an error that can be explicitly handled in the task model. Usually, these errors can be dealt with by trying the action again, or trying the action with a different controller.

However, when one of these underlying components fails, or when error handling is not sufficient, the system has no way of reasoning about error recovery. The top-down model becomes less reliable when the peg is bent out of position: at a 45 degree angle, many of the provided drop points will cause the ring to fall down and miss the peg due to gravity as shown in Figure 4. Since it treats the peg drop positions as a black box, it has no way of reasoning about how the environment has changed.

Bottom-up grounding achieves similar results using a very different strategy. There are a very large number of possible low-level symbols: in this case study, there are 36 objects in the world and 26 predicates. These objects are different components of the simulated WAM arms, peg stems and bases, the ring, and the stage the robots are attached to. This does not include any abstract entities such as grasp or drop points or movement waypoints which are used in the task.

There are a large number of low-level predicates necessary just to specify the spatial relationships between these 36 objects, including six one-dimensional geometric predicates: `higher_than`, `left_of`, `right_of`, `lower_than`, `in_front_of`, and `behind_from`. Each of these corresponds to a difference in x, y, or z axis relative to one the center of one of the 36 rigid objects in the world, describing the relationship in one dimension between two objects from a frame of reference. With $n = 36$

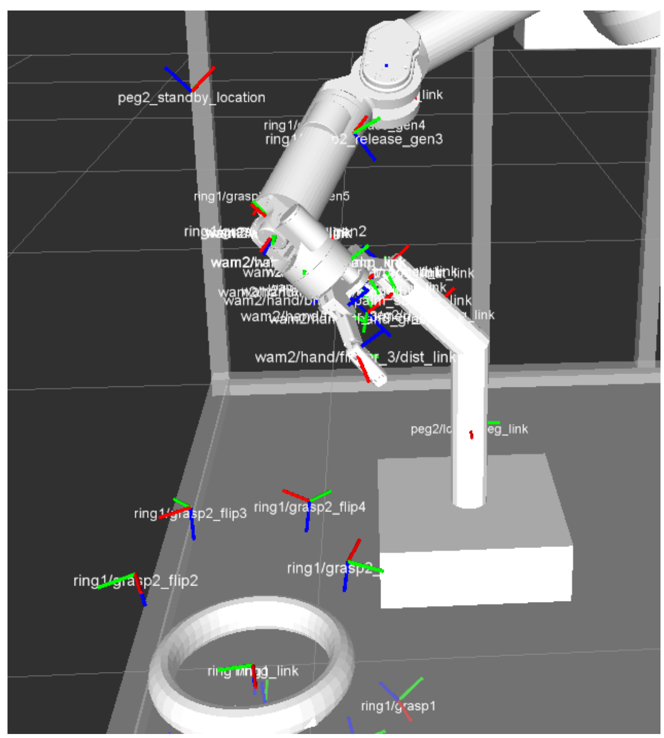

Fig. 4: Many of the same rules for placing the peg no longer work when it is bent at a 45 degree angle.

objects as possible reference frames, we have $n^2(n-1) = 45360$ possible assignments to each of these predicates. We also consider the often-used but poorly-defined predicates `nearby` and `nearby_mesh`, comparing distances between centers or meshes of two objects, respectively.

Our bottom-up implementation shows how using these low-level predicates can produce unintended effects; the movements selected by the system are often very dissimilar to those that a human might select. In addition, concepts which were easy to specify in the top-down model – such as movement waypoints – are much more complex expressions. For example, to pick up the ring from the first peg, the arm moves to a standby position before heading to a grasp point. This position should be behind and above the peg, near the side of the ring the arm is reaching for. This means that the predicates `left_of[arm1, ring]`, `left_of[arm1, peg1]`, `higher_that[arm1, ring1]`, and `behind_from[arm1, peg1]` all need to be true, and the set of collision predicates all need to be false.

Neither the top-down or bottom-up task models fully capture the requirements of the task. The top-down model relies on low-level actions with STRIPS-like properties to complete a task and inform the higher-level model whether they have succeeded or failed. On the other hand, the bottom-up model relies on a large set of predicates that approximate, but do not exactly capture, task requirements. For example, the top-down model will return successfully even when it has dropped the ring in the wrong position, because all of its component sub-tasks executed successfully. The bottom-up model does not plan to make sure the ring does not get caught on the first peg, but does get caught on the second one, because its plan is based only on spatial primitives. Effect-based explicit grounding addresses these limitations by computing certain properties with the Gazebo physics engine.

Our case study demonstrates the usefulness of different strategies when solving different problems. Grasp points were specified by human demonstration, and are therefore examples of top-down grounding. However, we stress that without enumerating the physical conditions that must be true for an action to have been successful, there are corner cases which cause both of the implicitly grounded models to fail.

V. Conclusions

We examined the challenges in appropriately grounding semantic predicates in task ontologies: predicates should be represented in such a way that they encode real-world, task-motivated values, rather than tokenized output encoding human-centric concepts. At the same time, these predicates need to be relatable to the high-level goals which may be specified by a human. High level-predicates specified by humans are likely to break down when moved into another task or domain, as they are designed with a specific task in mind and not grounded in the physics of the world or the capabilities of the robot. Additionally, since as more objects are introduced into the world, the number of possible spatial relationships an action can attempt to satisfy grows.

Our experiments showed three different ways of specifying the automated performance of a straightforward collaborative robotic task. While it would be ideal for the robot to have an intrinsic understanding of the effects all of its physical actions may have on the environment, this may be impractical; instead, any useful task specification would likely include a mix of low- and high-level symbols with explicit grounding used selectively to check for errors.

An explicit grounding may allow us to check the accuracy of a model, but is not feasible for every step of a task due to computation costs. Instead, we suggest learning bottom-up models for tasks that *implicitly* model the world in a way that allows us to reason about it. Then, an explicit grounding of the world can be used to check and improve these bottom-up models, adding additional rules as necessary.

References

[1] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 335–346, 1990.

[2] J. Y. Lee, P. Mucksavage, C. P. Sundaram, and E. M. McDougall, "Best practices for robotic surgery training and credentialing," *The Journal of urology*, vol. 185, no. 4, pp. 1191–1197, 2011.

[3] J. Wolfe, B. Marthi, and S. J. Russell, "Combined task and motion planning for mobile manipulation." in *ICAPS*, 2010, pp. 254–258.

[4] R. Philippsen, N. Nejati, and L. Sentis, "Bridging the gap between semantic planning and continuous control for mobile manipulation using a graph-based world representation," in *proceedings of the workshop on hybrid control of autonomous systems, Pasadena*, 2009.

[5] M. Tenorth and M. Beetz, "Knowrob: A knowledge processing infrastructure for cognition-enabled robots," *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013.

[6] C. McGann, F. Py, K. Rajan, H. Thomas, R. Henthorn, and R. McEwen, "A deliberative architecture for auv control," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 1049–1054.

[7] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *AAAI*, 2008, pp. 1433 – 1438.

[8] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, vol. 1, 2012, pp. 41 – 48.

[9] K. Welke, P. Kaiser, A. Kozlov, N. Adermann, T. Asfour, M. Lewis, and M. Steedman, "Grounded spatial symbols for task planning based on experience," in *13th International Conference on Humanoid Robots (Humanoids). IEEE/RAS*, 2013.

[10] C. McGann, E. Berger, J. Bohren, S. Chitta, B. Gerkey, S. Glaser, B. Marthi, W. Meeussen, T. Pratkanis, E. Marder-Eppstein, *et al.*, "Model-based, hierarchical control of a mobile manipulation platform," in *Proc. of ICAPS Workshop on Planning and Plan Execution for Real-World Systems*, 2009.

[11] N. Dantam, P. Koine, and M. Stilman, "The motion grammar for physical human-robot games," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5463–5469.

[12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[13] J. Bohren and S. Cousins, "The smach high-level executive [ros news]," *Robotics & Automation Magazine, IEEE*, vol. 17, no. 4, pp. 18–20, 2010.

[14] S. Stock, M. Günther, and J. Hertzberg, "Generating and executing hierarchical mobile manipulation plans," in *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. VDE, 2014, pp. 1–6.

A Method for designing Dialogue Systems by using Ontologies

Ewerton José Wantroba
Department of Computer Science
Institute of Mathematics and Computer Sciences
USP - University of São Paulo, Brazil
Email: wantroba@icmc.usp.br

Roseli Ap. Francelin Romero
Department of Computer Science
Institute of Mathematics and Computer Sciences
USP - University of São Paulo, Brazil
Email: rafrance@icmc.usp.br

***Abstract*—The use of ontologies in Computer Science has as main purpose to allow sharing and reusing of knowledge. It is important that the concepts present in the ontology have in fact a formal specification. This article presents a methodology for designing Dialogue Systems, turning easier the task of building the knowledge base for a dialogue system. A method to design and aggregate existing ontologies in these systems are also being proposed. For this, pattern matching, state of art natural language processing tools, thesaurus and the language AIML are also used. The proposed approach has been applied to a recepcionist system and it will be shown through the several experiments performed a satisfactory performance of the proposed dialogue system.**

I. Introduction

An ontology is a knowledge structure used to represent (formally) and share domain knowledge through modeling and establishing a framework about relevant concepts and the semantic relationships among these concepts [1], [2] and [3].

Ontology structures explicitly represent domain knowledge by using a format comprehensible by a machine. They can be incorporated into computer applications and systems, facilitating the annotation data [4], the decision making process [5], information retrieval and natural language processing [6]. In addition, they serve as part of the Semantic Web [7]. Ontologies have also the potential to support the process of clinical decision making (CDS), increasing the reuse of data and knowledge systems [8].

Ontology development, according to the principles of the developing ontology, can potentially facilitate interoperability and reuse. On the other hand, designing ad hoc ontologies, without the use of development standards, have created an environment in which there are numerous ontologies with limited ability for both communicate amongst themselves and the reuse of the knowledge [1], [9] and [10].

Although there is no consensus on how to develop ontologies, several approaches have been described to better share some common elements of development. Likewise, while formal evaluation methods have the potential to maximize the benefits of ontologies within the areas of computer science, philosophy and life, there is no standard approach to assess the quality of ontologies, from the perspective of their intrinsic or extrinsic value characteristics (ie, the usefulness of a particular task) [11], [12], [13].

This paper presents an architecture for dialog system which includes a methodology for building dialogue using ontologies. In Section II is described the complete architecture for the dialog system, wherein the dialogue module which makes use of ontologies is inserted. In Section III is explained, in details, the methodology proposed. Then the inference engine consisting of two main methods making use of natural language processing tools along with the thesaurus WordNet is presented in Section IV. For validating the proposed methodology was applied for a receptionist system capable to answer questions about information relatet to Institute of Mathematics and Computer Sciences - ICMC-USP and the results are presented in Section V.

II. Architecture

It has been proposed and developed an architecture for a dialogue system (Figure 1). The proposed architecture is independent of the domain of language and can be used in any dialogue system. The advantages of its use include the easy inclusion and updating of knowledge, represented in the AIML files format [14] in this work, but the informatioc can be stored using other ways (such as a database). The inference engine mechanism, presented in Section IV), is suitable to use for receptionists systems, but can also be adapted to other dialogue systems.

The architecture is composed by four modules. The first one is the interface module for establishing the communication between the user and the desktop avatar or browser avatar, the lattest allows that people all over of world accessing the recepcionist system. The second is Web Module for providing the encoding and decoding of information received by the interface module. The third one is the voice module allowing voice and textual interaction and enables the system to recognize and synthetize the voice. It is optional, due to its complexity, but highly recommended, as the benefit of its use in relation to a system that allows only textual interaction. The fourth module is the dialogue module, that is responsible for storing the knowledge base in AIML file format [14] and running the inference engine (Section IV). It is capable for holding multiple users simultaneously connected with Avatar Valerie, without regard to conflict of information.

Based on the knowledge representation model presented in Section [15], responsible by the inferences about the information, the dialog module contains the following steps:

1) *parsing*: to verify if the matching of word by word occurs, using the algorithm of *Matching Behavior*

implemented by the parser available at Language AIML [14]. In this work, we have adopted the parser ProgramD (http://aitools.org/Programd) due to the fact that it facilitates the web communication.

2) *semantics*: instead of producing semantic trees, a XML tree is constructed through the own structure of language AIML.
3) *knowledge base and structures of the real world*: they are inserted by the developer of < category > 's AIML, which contains information of the language, structure of questions and answers external database.

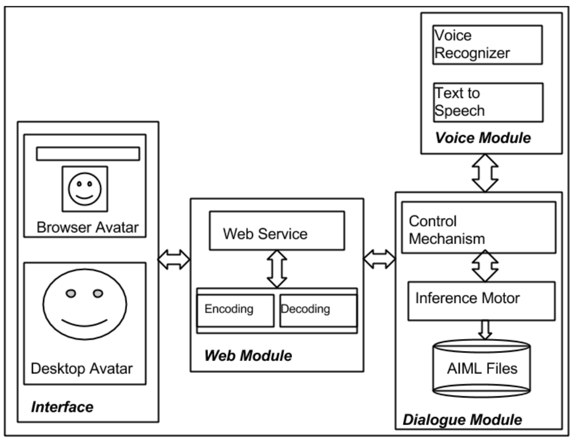

Fig. 1. System Architecture.

III. Building the Knowledge Base using Ontologies

The most important task that a chatterbot have to perform is to be able to communicate with humans through natural language. The method of knowledge construction and organization required for this task is of paramount importance in orderto the chatterbot can be acceptable. The choice of method will directly affect the system's functionality and limitations. In this section, we will explain how the knowledge base has been created by using ontologies.

For the construction and organization of the dialogue in Portuguese referring to the Institute of Mathematics and Computer Sciences (ICMC-USP), it has been developed and used the methodology explained to follow. Structured data used are provided with a database *MySql* provided by ICMC-USP. Further, unstructured information are also used obtained through *FeedsRSS*, pages of ICMC-USP and such data first go through a process of structuring data as it follows:

1) Build the Ontologies:
   a) *Build the Classes* from the structured data contained in database, such as, Name, Room, E-mail,...).
   b) *Build the Generalizations "is-a"* related to the classes to their respective superclasses. (i.e. EMail is an address; Classroom is a local;). How much more generalizations are created at this stage richer domain knowledge will be. Another important fact is about the generalizations which can be changed later without to affect the construction of the dialogue afterwards. All classes and generalizations known by the receptionist developed system are shown in Figure 2.

Fig. 2. The classes and generalizations of the ontology.

   c) *Build the Object Properties and Data Properties*, such properties represent relationships between individuals of the ontology. Each property has:
      i) *Name:* usually a verb. (Ex. studies)
      ii) *Domain:* classes of individuals belonging to the domain (eg. Professor)
      iii) *Range:* classes of individuals belonging to the range (eg. Interest Area). If it is a Date Property then these values are considered as literal values and do not classes, eg. [Seminar] isDescribed**string**)
      iv) *characteristics: they can be of the following types:*
         A) *Functional*: If a property is functional for a particular **individual a**, there may be up to one **individual b** that is related to **a** through that individual property;
         B) *Inverse functional*: If a property is an inverse functional, this means that the functional property is its inverse. For the **individual a** can exist at most one related to **a** through individual property individual;
         C) *Transitive*: If a **P** transitive property relates the **individual "a"** to **individual "b"**, and also a **individual "b"** to **individual "c"**, it is inferred that the **individual**

"a" is related to the **individual "c"** according to **P**;

D) *Symmetric*: If a **P** property is symmetric, and relates a **individual "a"** to **individual "b"**, then the **individual "b"** is also related to the **individual "a"** through the property **P**.

In Figure 3 it is shown some properties created for the ontology developed in this work.

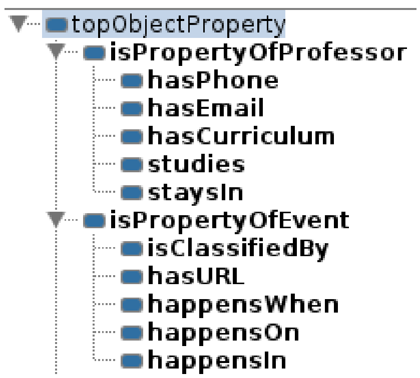

Fig. 3. Some properties of the ontology.

d) **Create the inverse properties:** for each property built before, its inverse must be created. Examples:
1. $[Professor]$**studies**$[InterestArea] \rightarrow [InterestArea]$**isStudiedBy**$[Professor]$;
2. $[Author]$**presents**$[Seminar] \rightarrow [Seminar]$**isPresentedBy**$[Author]$.
Created all inverse properties, the inference mechanism of ontology (*Reasoner*) will be able to infer all the properties automatically, including reverse ones.

e) **Build the individual of the ontology** with information from the data sources (Database, Feeds, Sites, ...).
Proposal for creation of individuals with compound names:
In the proposed approach, an individual is identified by its full name. For example, there is an individual with the name "Roseli Aparecida Francelin Romero", its name will be: Roseli_Aparecida_Francelin_Romero, being single and identified by this. But there is no need that the user enter the full name of any individual. This is possible thanks to the use of a class called "Key", which contains individuals with only a single word such as name and identifier. In that case, there would be four individuals (Roseli, Aparecida, Francelin and Romero), related to the teacher. So it is only necessary the user types only one of these names and the system will able to know about what individual the user is talking about. In case of ambiguity, for example, there are more than one person named Roseli, the system will need more information about the individual name and if the user has not typed or spoken about it, the system will require.

f) **Relating the individuals created by using the Properties** (Object and Data Properties) (as it was mentioned, the inverses are inserted automatically and not necessarily need be expressed at this stage).

2) **Design patterns will match with user inputs and the corresponding answers in natural language**. The AIML language 's formalism were used at this stage. A category is created for each input pattern and AIML file contains as many categories as are the properties of the ontology. Each AIML category contains:

a) **Pattern:** It is created a pattern for each Property of the ontology. An example of input pattern is presented to follow:
$Property[Professor]studies[InterestArea]$.
$< pattern >$
$PROFESSOR * studies$
$< /pattern >$

The reverse would be automatically constructed:
Inverse property:[InterestArea] isStudiedBy [Professor]
$< pattern >$
$InterestArea * isStudiedBy$
$< /pattern >$

b) **Template:** Each *template* contains the answer format providing information to the user in natural language based on ontology. Example:
$< template >$
$Theprofessor * studiesthoseInterestAreas :$
$< ontologia.Inference(Propriedade = studies; Individual = * >$
$< /template >$
(The inference above returns all the Interest Areas of the "Professor *")

Note: A mechanism automatically creates all the patterns based on the ontology properties. It will be the developer responsibility to write the *template* for all the patterns. Otherwise, the system will only display the data as a response.

The proposed method for establishing the dialogue to turn the system able to act as a receptionist at ICMC-USP will be presented in the next section.

```
<property name="name" value="Valerie"/>
<property name="gender" value="female"/>
<property name="sign" value="Capricorn"/>
<property name="city" value="São Carlos"/>
<property name="footballteam" value="Brazil"/>
<property name="favoriteactor" value="Tom Hanks"/>
<property name="hair" value="I have some plastic wires"/>
<property name="job" value="chat bot"/>
<property name="wear" value="my usual plastic computer wardrobe"/>
```

Fig. 4. Stretch of the configuration file of Personality Avatar Valerie.

IV. INFERENCE ENGINE

Algorithm 1 Inference engine.

Require: Ontologies and WordNet, Models PLN (Sentences detector, tokenizer, POS Tagger) and User Input
{Uses a model of detector sentences}
$sentence \leftarrow firstSentences(userInput)$;
{uses a tokenizer to separate words}
$words \leftarrow tokenizer(sentence)$;
{Part of Speech tagger classifies words according to their grammatical classes}
$grammaticalClasses \leftarrow posTagger(words)$;
while all words are not analyzed **do**
  if word is a noun **then**
    $synonymous \leftarrow returnSynonymous(word)$;
    {the WordNet returns all similar words}
    $candidateWords \leftarrow word + synonymous$;
    {checks whether the words are classes in the ontology}
    $candidateClasses \leftarrow returnClasses(candidateWords)$;
    {Search in the ontology if the words are individuals of classes}
    $candidateIndividuals \leftarrow returnIndividuals(candidateWords)$;
    {Classes of individuals candidates}
    $ClassesCandidateIndividuals \leftarrow returnClasses(candidateIndividuals)$;
  end if
end while
{call the algorithm 2 for finding the correct AIML pattern based on candidateClasses, candidateIndividuals, ClassesCandidateIndividuals}
$pattern \leftarrow findPattern(...)$;
$template \leftarrow retornaTemplate(pattern)$;
return template

Algorithm 2 Finding the AIML pattern that contains the response correspondent to the users input, based on Classes and Individuals already elucidated.

Require: Classes candidateClasses,Individuals candidateIndividuals, Classes ClassesCandidateIndividuals
$candidateIndividuals \leftarrow desambigua(candidateIndividuals)$;
{disambiguation of individuals by keyword}
{find all Super Classes of individuals candidates}
$SuperClassesCandidateIndividuals \leftarrow returnSuperClasses(ClassesCandidateIndividuals)$;
{find all Sub Classes of individuals candidates}
$SubClassesCandidateIndividuals \leftarrow returnSubClasses(candidateClasses)$;
{Only if there is an individual then it is the individual about whom you want to obtain information}
if candidateIndividuals.size == 1 **then**
  $individual \leftarrow candidateIndividuals[0]$;
  $individualClasse \leftarrow returnClasse(individual)$;
  {performs permutation among individual, candidateClasses, SuperClassesCandidateIndividuals, SubClassesCandidateIndividuals}
  $permutations \leftarrow permutation()$;
  $properties \leftarrow findProperties(permutations)$;
  {if there be any among the permutation property then there is no matching}
  if properties.size == 0 **then**
    return ”No Match”;
  end if
  {there is more than one property, then the user needs to choose which among them}
  if properties.size > 1 **then**
    return ”There are more than one Match”;
  end if
  {There is one Match}
  if properties.size == 1 **then**
    $propertie \leftarrow properties[0]$;
    $classeRange \leftarrow properties.getRange()$;
    $AIMLpattern \leftarrow < pattern > individualClasse + propertie + classeRange < /pattern >$;
    return AIMLpattern;
  end if
end if
{if there is more than one person then the user must resolve the ambiguity}
if candidateIndividuals.size > 1 **then**
  return ”Ambiguity”;
end if

In order a dialogue contained in AIML file can be inferred by a robot, it is necessary to use interpreters, which are tools that can be found in (http://aiml-programr.rubyforge.org), such as, Program P (http://alicebot.sweb.cz/) and Program D (http://aitools.org/Programd) among others. We have opted to use the interpreter Program D, which is able to maintain a dialogue with multiple users simultaneously and futher to facilitate its use by a Web service.

For interaction in English language, all the knowledge base provided by ALICE *chatterbot* was used. Its creator [14] proposed that the personality of a chatterbot, developed using AIML, is easily transcribed by defining variables called properties. These properties are presented in a configuration file. The Avatar Valerie contains 75 properties. In Figure 4, it is shown a part of the file that defines the personality variables of Avatar.

Once added all knowledge of A.L.I.C.E. Avatar into Valerie system, the matching of patterns among the entries in English is given by *Matching Behavior* Algorithm contained in Program D interpreter.

The interaction in Portuguese language allows to the user to get the information regarding the ICMC-USP, based on the detailed proposal in Section III. For the matching of input patterns with knowledge base of *chatterbot*, Valerie, was proposed the algorithm 1 presented to follow.

The first step of Algorithm 1 consists of pre-processing the user's input. Then is obtained and used only the first sentence, being that each word is separated and classified by *tokenizer*, and by *Part of Speech tagger*. After, every word is analyzed and the nouns are selected. These will be the candidate words to be individuals of the ontology classes. The goal at this point is to find the AIML pattern that will contain the answer to a presented question. The details are stored as AIML patterns and are presented in Section III. The method to find the pattern matching between the Classes and individuals, discovered in the phrase, is detailed in Algorithm 2 as it follows.

It is assumed that the user will always be asking a question about an individual present in the ontology. The answer is contained in a property of the concerned individual. Then, we need to find out what is the property that contains the answer.

This, in algorithm 2, is basically a permutation among the class of the individual candidate, their super classes, sub classes and other classes found in the standard input. The goal is to find a ontology property that contains the answer. It is noteworthy that the method to disambiguate individuals was presented in Section III.

V. Experiments and Results

For validating the proposed system, a questionary was applied to ICMC-USP community, which speaks brazilian portuguese. It contains five scenarios of interaction with the system, in which the user should answer some questions.

A total of 15 questions were suggested to the user among the five scenarios, in which the user should ask to the system (the Avatar) to get the information. For each question, the user should answer if the system returned the information required by ticking one of three possible answers: ”Yes, the system hits the first attempt”; ”Yes, but the system hits after several attempts”; and ”No, the system did not respond”.

The system accuracy obtained was 70% having the Avatar answered correct on average 42% of the questions on the first attempt and 28% after a few tries. On average, the system was not able to answer only in 30% of the cases, which usually involved atypical spelling of proper names. This results are showed in Figure 5.

Noteworthy the questions that got the best and the worst accuracy rate. The system performed better on the questions about a Seminar entitled ”Allocation Of Tasks And Communication For The Coordination Of Robots”, responding to 70% of the questions on the first attempt, 20% after a few tries and 10% the system did not respond. Also, it presented 70% accuracy on the first attempt in question about the teacher called ”Roseli Aparecida Francelin Romero”, with 10% accuracy after some trial and not responding in 20% of the tests. In Figure 6, it is shown the accuracy of the proposed system answering questions that not include unusual proper names.

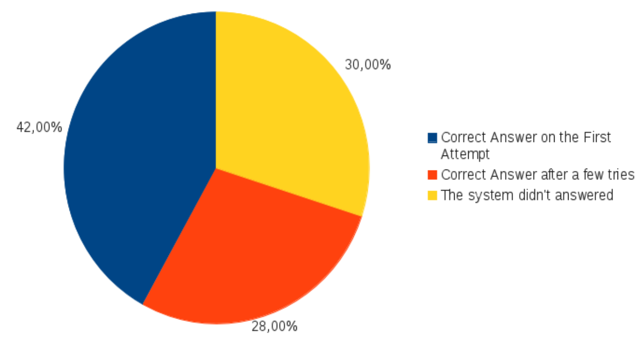

Fig. 5. The System Accuracy.

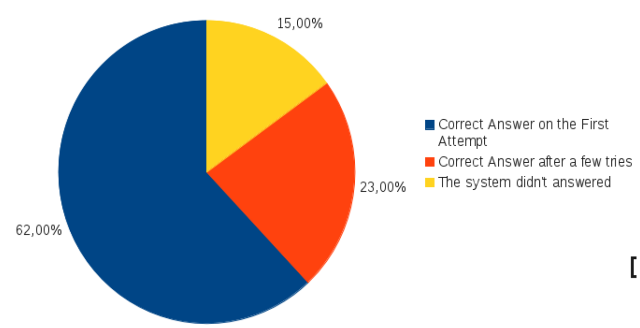

Fig. 6. The System Accuracy with questions without the presence unusual proper names.

The worst performance was in question about a Seminar presented by ”Fabio Ruffino”, where the system failed to respond in 70% of the cases, and hitting the 30% of the tests on the first attempt, this being the only one in which the system does not respond to more than 50% of the tests. This is due to the name ”Ruffino” with the two letter ”f” is not usual in Brazil. In Figure 7, it is shown the system accuracy responding questions containing unusual proper names.

VI. Conclusion

In this work, a method was proposed for the dialog construction based on ontologies for incorporating in an avatar to turn it able to act as a recepcionist system. This proposal can be easily extended to other types of systems. This methodology was applied and tested in the context of a recepcionist for ICMC-USP and presented a satisfactory accuracy. The use of methodologies for building such a system, and in conjunction with the proposed pattern for matching of the input and output information allowed to the system to be able to answer questions based on an ontology. This system will be used

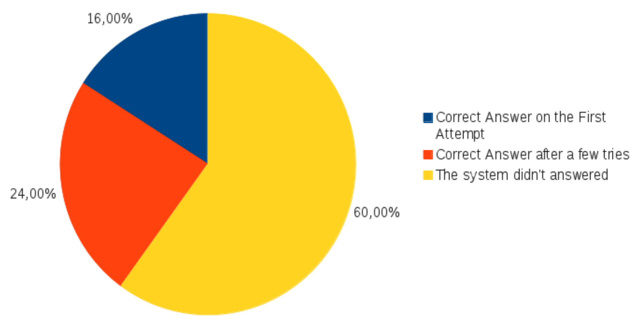

Fig. 7. The System Accuracy with questions containing unusual proper names.

as a recepcionist at ICMC-USP, providing information about teachers, seminars and other items of the user interest. The testing showed that system has a high accuracy on questions in which key words (nouns) are easily recognized by the speech recognizer (e.g. the questions about Seminars and about the teacher names, and a low accuracy in questions containing unusual proper names.

Thanks to the use of ontologies, there is a possibility of building an Avatar with its own personality and reusing the knowledge of A.L.I.C.E. This is possible due to easy inclusion of properties separately from knowledge base itself. Originally the software A.L.I.C.E. only allowed interaction by text, and thus, the use of their knowledge base on any system capable of adding features as voice and facial expressions, turns it to be more attractive, intelligent and easy to use.

The availability of the dialogue system in the *Web* provided an easier the interaction with the user. Considering that the development of microarray chatterbots is a cyclical process in which each time the system is improved based on the previous interactions, the greater the range of individuals, more suitable the system will become.

As future work, we intend to increase the intelligibility of the system. One way to improve the dialogue is to increase the complexity of the ontology and hence adding more complex properties for the search. It is noteworthy that to do this, it is necessary an improvement in the stage of preprocessing user input, with improved techniques for Natural Language Processing. We also intend to use, develop and incorporate techniques allowing the detection of repeated questions by the user and alert him about them.

Acknowledgment

The authors would like to thank Carnegie Mellon University for grant AVATAR Valerie and CNPq and FAPESP for financial support, which enabled the realization of this work.

References

[1] O. Bodenreider and R. Stevens, "Bio-ontologies: current trends and future directions," *Brief Bioinform*, vol. 7, pp. 256–274, 2006.

[2] N. F. Noy and D. L. Mcguinness, "Ontology development 101: A guide to creating your first ontology," Tech. Rep., 2001.

[3] M. Uschold, M. Gruninger, M. Uschold, and M. Gruninger, "Ontologies: Principles, methods and applications," *Knowledge Engineering Review*, vol. 11, pp. 93–136, 1996.

[4] N. Shadbolt, W. Hall, and T. Berners-Lee, "The semantic web revisited," *Intelligent Systems, IEEE*, vol. 21, no. 3, pp. 96–101, Jan 2006.

[5] M. A. Musen, "Scalable software architectures for decision support," *Methods of information in medicine*, vol. 38, no. 4/5, pp. 229–238, 1999.

[6] D. L. Rubin, S. E. Lewis, C. J. Mungall, S. Misra, M. Westerfield, M. Ashburner, I. Sim, C. G. Chute, M.-A. Storey, B. Smith *et al.*, "National center for biomedical ontology: advancing biomedicine through structured organization of scientific knowledge," *Omics: a journal of integrative biology*, vol. 10, no. 2, pp. 185–198, 2006.

[7] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[8] T. J. Bright, E. Yoko Furuya, G. J. Kuperman, J. J. Cimino, and S. Bakken, "Development and evaluation of an ontology for guiding appropriate antibiotic prescribing," *J. of Biomedical Informatics*, vol. 45, no. 1, pp. 120–128, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jbi.2011.10.001

[9] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall *et al.*, "The obo foundry: coordinated evolution of ontologies to support biomedical data integration," *Nature biotechnology*, vol. 25, no. 11, pp. 1251–1255, 2007.

[10] L. N. Soldatova and R. D. King, "Are the current ontologies in biology good ontologies?" *Nature biotechnology*, vol. 23, no. 9, pp. 1095–1098, 2005.

[11] A. C. Yu, "Methods in biomedical ontology," *J. of Biomedical Informatics*, vol. 39, no. 3, pp. 252–266, Jun. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.jbi.2005.11.006

[12] J. Brank, M. Grobelnik, and D. Mladenić, "A survey of ontology evaluation techniques," 2005.

[13] N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. Storey, C. G. Chute *et al.*, "Bioportal: ontologies and integrated data resources at the click of a mouse," *Nucleic acids research*, vol. 37, no. suppl 2, pp. W170–W173, 2009.

[14] S. Wallace, R, "The anatomy of a.l.i.c.e." *A.L.I.C.E. Artificial Intelligence Foundation, Inc.*, 2004.

[15] G. F. Luger, *Inteligncia Artificial: estruturas e estratgias para a soluo de problemas complexos*. Brooknann, 2004.