Performance Evaluation of Robotic Knowledge Representation (PERK)

Craig Schlenoff[a,b]
[a]National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899, USA
craig.schlenoff@nist.gov

Sebti Foufou[b,c]
[b]University of Burgundy,
LE2i Lab., Dijon, France.
sfoufou@u-bourgogne.fr
and
[c]Computer Science and Eng.
Qatar University, Doha, Qatar
sfoufou@qu.edu.qa

Stephen Balakirsky[a]
[a]National Institute of Standards and Technology
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899,USA
stephen.balakirsky@nist.gov

ABSTRACT

In this paper, we explore some ways in which symbolic knowledge representations have been evaluated in the past and provide some thoughts on what should be considered when applying and evaluating these types of knowledge representations for real-time robotics applications. The emphasis of this paper is that the robotic applications require real-time access to information, which has not been one of the aspects measured in traditional symbolic representation evaluation approaches.

Categories and Subject Descriptors

I.2.4. **[Computer Methodologies]:** Artificial Intelligence: Knowledge Representation Formalisms and Methods – Representation languages

General Terms

Measurement, Performance

Keywords

Robotics, knowledge representation, performance metrics, real-time, ontologies

1. INTRODUCTION

A robot can only perform tasks based on what it knows, which is often captured within the robot's internal knowledge representation. This representation can take many forms and knowledge can be captured at various levels of specificity. With the growing complexity of behaviors that robots are expected to perform, the need to measure the knowledge representation, in terms of coverage, the ability to reason to infer new knowledge, and the ability to successfully complete complex tasks, is becoming more evident.

Knowledge representations have historically been evaluated using metrics such as completeness (Is all necessary knowledge represented?), expressiveness (Can all necessary knowledge be represented?), accuracy (Is the represented knowledge correct?), and consistency (Are there contradictory facts represented?)[1, 2]. While these metrics are important in a theoretical sense, knowledge representation for robotics introduces a series of additional metrics, such as performance (real-time access), flexibility (ability to constantly update knowledge as new information becomes available), and relevance (is information represented at a level of resolution that can be used by planning systems). In addition, the way that the representations are evaluated must change when introducing these new metrics. For example, while running a consistency checker can help to identify contradictory knowledge, it does not assess the representation's ability to respond to an ever-changing environment. Successful measures for these types of metrics may include the ability (and time) to answer what-if questions, the ability to support real-time planning, etc.

In this paper, we explore some ways in which knowledge representations have been evaluated in the past and provide some thoughts on what should be considered when evaluating knowledge representation for real-time robotics applications. This paper is organized as follows:

- Section 2 discusses current knowledge representation approaches in the robotics domain
- Section 3 describes an ontology standardization effort that will serve as the basis for future research efforts
- Section 4 describes some previous efforts that have explored how to measure the performance of symbolic knowledge representations with an emphasis on ontologies
- Section 5 attempts to categorize the types of metrics that have been used in the past along with some thoughts on their applicability to the robotics domain
- Section 6 concludes the paper by discussing the relationship between ontology metrics and traditional robotics knowledge representation approaches and where the current gaps lie.

2. KNOWLEDGE REPRESENTATIONS FOR ROBOTICS

Traditionally, robots use a wide array of knowledge representations. Some of these include parametric knowledge, spatial knowledge, and symbolic knowledge. A good overview of these types of knowledge and how they have been applied

PerMIS'12, March 20-22, 2012, College Park, MD, USA.

to the robotics domain can be found in [3]. An overview of these types of knowledge is described below.

2.1 Parametric Knowledge

The lowest levels of any control system, whether for an autonomous robot, a machine tool, or a refinery, are at the servo level, where knowledge of the value of system parameters is needed to provide position and/or velocity and/or torque control of each degree of freedom by appropriate voltages sent to a motor or a hydraulic servo valve. The control loops at this level can generally be analyzed with classical techniques and the "knowledge" embedded in the world model is the specification of the system functional blocks, the set of gains and filters that define the servo controls for a specific actuator, and the current value of relevant state variables. These are generally called the system parameters, so we refer to knowledge at this level as parametric knowledge.

Figure 1shows a traditional PD (Proportional Derivative) servo control for a motor of a robot arm. All six or seven motors that drive the arm will have basically the same servo control, but each will have different parameters because there are different size motors driving different loads at different points in the arm. Any errors that deal with a single degree of freedom, such as ball screw lead errors, contact instabilities, stiction, and friction are best compensated for at this level.

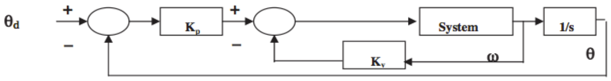


Figure 1: PD Servo Control

2.2 Spatial Knowledge

Above the servo level are a series of control loops that coordinate the individual servos and that require what can be generally called "geometric knowledge," "iconic knowledge," "metrical maps," or "patterns." This knowledge is spatial in nature and can be defined as 2D or 3D array data in which the dimensions of the array correspond to dimensions in physical space. The value of each element of the array may be Boolean data or real number data representing a physical property such as light intensity, color, altitude, range, or density. Each element may also contain spatial or temporal gradients of intensity, color, range, or rate of motion. Each element may also contain a pointer to a geometric entity (such as an edge, vertex, surface, or object) to which the pixel belongs.

Examples of iconic knowledge include digital terrain maps, sensor images, models of the kinematics of the machines being controlled, and knowledge of the spatial geometry of parts or other objects that are sensed and with which the machine interacts in some way. This is where objects and their relationship in space and time are modeled in such a way as to represent and preserve those spatial and temporal relationships, as in a map, image, or trajectory.

For industrial robots, machine tools, and coordinate measuring machines, the first level above the servo level deals with the kinematics of the machine, relating the geometry of the different axes to allow coordinated control. Linear, circular and other interpolation and motion in world or tool coordinates is enabled by such coordination. The "knowledge" here may be the kinematic equations or Jacobian coefficients that define the geometric relationships of the axes, or the mathematical routines for interpolation or coordinate transformations. It is at this level that systematic multi-dimensional geometric errors such as non-orthogonality of axes of a machine tool are considered.

For mobile autonomous robots, there are two main categories of spatial knowledge representation that are useful. These are sometimes referred to as metrical maps in the literature. One captures what the sensors see (the view "out the windshield"). This may be two-dimensional images, as is the case for CCD (Charge Coupled Device) cameras, or three-dimensional images, in the case of range sensors such as LADARs (laser Detection and Ranging). Some mobile robots successfully accomplish their goals by planning based on a world model derived purely from the sensor image view.

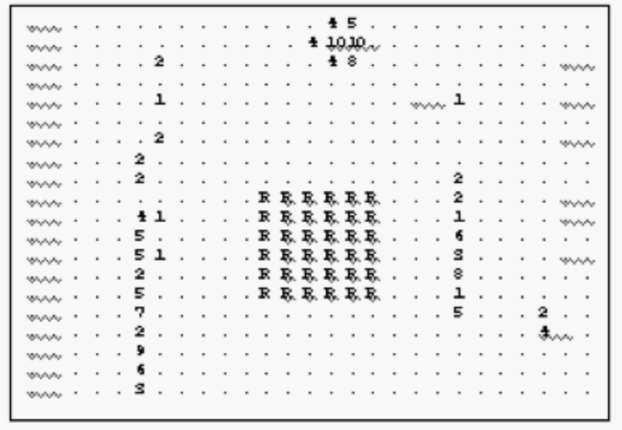


Figure 2: Occupancy Grid Map for Mobile Robot

Figure 2 shows a typical local map from a mobile robot navigating through an indoor environment. The robot's position at the center is indicated by marking the occupied cells with "R". The numbers in certain cells indicate the degree of confidence that there is an obstacle occupying that cell.

The second type of spatial representation is akin to the "bird's-eye-view." Figure 3 shows a higher level map for path planning for outdoor navigation. This map contains several feature layers, including elevation, vegetation, roads, buildings, and obstacles. Digital maps are a natural way of representing the environment for path planning and obstacle avoidance, and provide a very powerful mechanism for sensor fusion since the data from multiple sensors can be represented in a common format. Digital terrain maps are essentially two-dimensional grid structures that are referenced to some coordinate frame tied to the ground or earth. A map may have multiple layers that represent different "themes" or attributes at each grid element. For instance, there may be an elevation layer, a road layer, a hydrology layer, and an obstacle layer. The software can query if there is a road at grid location [x, y] and similarly query for other attributes at the same [x, y] coordinates.

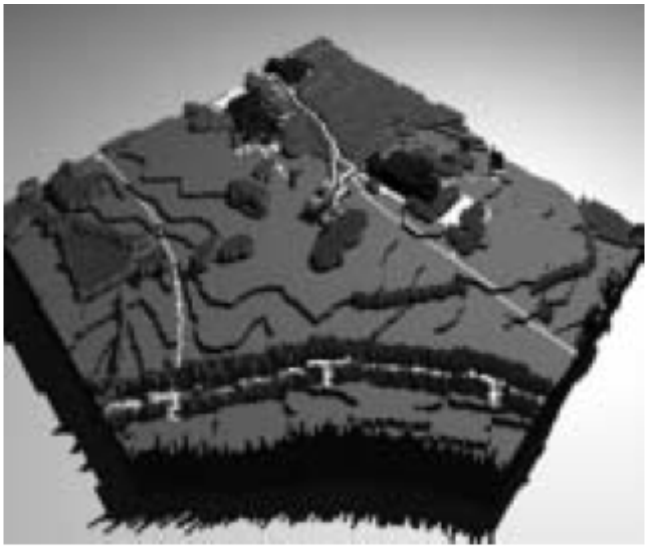


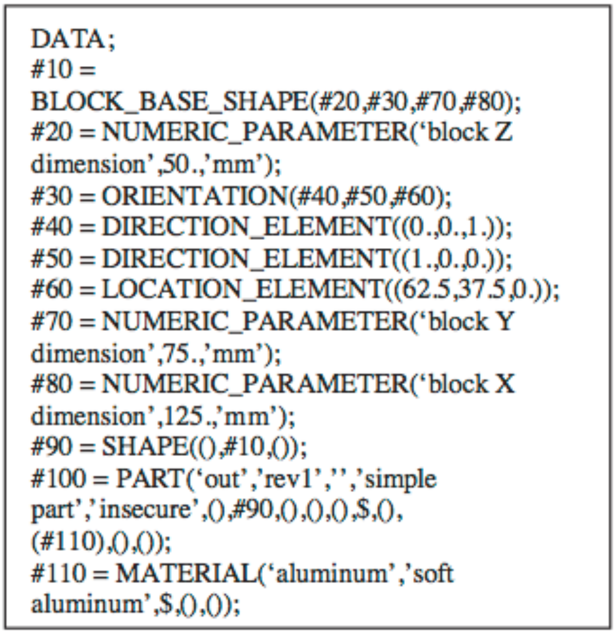
Figure 3: Multi-Terrain Digital Terrain Map

2.3 Symbolic Knowledge

At the highest levels of control, knowledge will be symbolic, whether dealing with actions or objects. It is at this level that a large body of relevant work exists in knowledge engineering for domains other than real-time control, such as formal logic systems or rule-based expert systems. Whether the knowledge is represented in terms of mathematical logic, rules, frames, or semantic nets, there is a formal linguistic structure for defining and manipulating and using the knowledge.

An example of a formal description of a solid model of a part is shown in Figure 4. A block is being described using International Standards Organization Standard for the Exchange of Product Model Data (STEP) Part 21 [4]. Note that this representation can be linked by pointers to a geometric representation where, for example, a block might be represented by equations of six planes with bounding curves and a coordinate transformation matrix to position the block within a given coordinate system.

Linguistic representations provide ways of expressing knowledge and relationships, and of manipulating knowledge, including the ability to address objects by property. Tying symbolic knowledge back into the geometric levels provides symbol grounding, thereby solving a serious problem inherent to purely symbolic knowledge representations. It also provides the valuable ability to identify objects from partial observations and then extrapolate facts or future behaviors from the symbolic knowledge. In the manufacturing domain, using a feature-based representation (which is symbolic) is reasonable at the generative planning level (Figure 5a). Graphical primitives (Figure 5b) that relate to the geometry can be tied to features to let users easily pick a feature (such as a pocket) by selecting on a portion of it on the screen. The geometric representation of each edge and surface that comprise a feature (Figure 5c) can be tied to the feature definition in order to facilitate calculations for generating the tool paths.

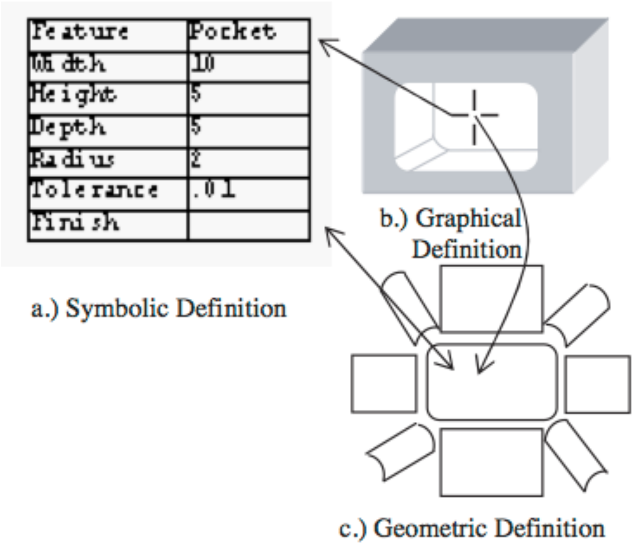


```
DATA;
#10 =
BLOCK_BASE_SHAPE(#20,#30,#70,#80);
#20 = NUMERIC_PARAMETER(‘block Z
dimension’,50.,’mm’);
#30 = ORIENTATION(#40,#50,#60);
#40 = DIRECTION_ELEMENT((0.,0.,1.));
#50 = DIRECTION_ELEMENT((1.,0.,0.));
#60 = LOCATION_ELEMENT((62.5,37.5,0.));
#70 = NUMERIC_PARAMETER(‘block Y
dimension’,75.,’mm’);
#80 = NUMERIC_PARAMETER(‘block X
dimension’,125.,’mm’);
#90 = SHAPE((),#10,());
#100 = PART(‘out’,’rev1’,’’,’simple
part’,’insecure’,(),#90,(),(),(),$,(),
(#110),(),());
#110 = MATERIAL(‘aluminum’,’soft
aluminum’,$,(),());
```

Figure 4: STEP Representation of a Block

Another type of symbolic representation for representing rules is ontological. Ontologies are definitions and organizations of classes of facts and formal rules for accessing and manipulating (and possibly extending) those facts. [5] There are two main approaches to creating ontologies, one emphasizing the organizational framework, with data entered into that framework, and the other emphasizing large scale data creation with relationships defined as needed to relate and use that data. Cyc [6] is an example of the latter, an effort to create a system capable of common sense, natural language understanding, and machine learning.

An ontology may be designed to make it easy for reasoning systems to reason using the ontology. This includes being able to infer information that may not be explicitly represented, as well as the ability to pose questions to the knowledge base and receive answers in return. One way of enabling this functionality is to represent the symbolic information in the world model in a logic-based, computer-interpretable format, such as in the Knowledge Interface Format (KIF) representation [7] and using a logic programming tool such as Prolog. [8]



Feature	Pocket
Width	10
Height	5
Depth	5
Radius	2
Tolerance	.01
Finish	

a.) Symbolic Definition

b.) Graphical Definition

c.) Geometric Definition

Figure 5: Pocket Feature

Through the use of an inference engine or theorem prover, information represented in this format could be queried, and logically-proven answers could be returned. As an example, a manufacturer may want to know whether a given set of fixture positions is suitable to fully inspect a part. Assuming that the necessary inspection points, access volumes, and machine capabilities are represented in KIF, the manufacturer could enter in the fixture positions and the system could logically-prove whether those positions are sufficient to fully inspect the part.

The focus of the remainder of this paper will be on symbolic representation, as it will be the focus of future research efforts.

3. IEEE ROBOTICS AND AUTOMATION SOCIETY (RAS) ONTOLOGIES FOR ROBOTICS AND AUTOMATION (ORA) WORKING GROUP

For the research effort described later in this paper, a standard knowledge representation (ontology) is needed. IEEE had formed a working group to explore the development of a standard robot ontology. It is anticipated that this ontology will serve as the basis for this work and is described below.

In October 2011, IEEE approved a new working group called Ontologies for Robotics and Automation (ORA) [9]. The goal of this working group is to develop a standard 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. The standard provides a unified way of representing knowledge and provides a common set of terms and definitions, allowing for unambiguous knowledge transfer among any group of humans, robots, and other artificial systems. To date, the working group is made up of over 115 members containing a cross-section of industry, academia, and government and representing over twenty countries.

The working group defines an ontology as a knowledge representation approach that represents key concepts, their properties, their relationships, and their rules and constraints. [10] 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.

The working group acknowledges that it would be extremely difficult to develop an ontology that could cover the entire space of robotics and automation. As such, the working group is structured in such a way as to take bottom-up and top-down approaches to addressing this broad domain. From a top-down approach, a sub-group entitled "Upper Ontology/Methodology". (UpOM) is exploring the identification or development of an upper ontology on which to hang more detailed concepts. In addition to this upper ontology, a methodology is being developed that would allow interested colleagues to propose additional concepts and reconcile any differences between the new concepts and those that already exist.

From a bottom-up perspective, three sub-groups have been formed which will take a detailed look at three sub-domains in the robotics and automation area. Those sub-domains are Autonomous Robots (AuR), Service Robots (SeR), and Industrial Robots (InR). Each of those subgroups will deeply explore their respective areas by identifying key concepts, along with their definitions, that need to be represented. The group's structure is shown in Figure 6. These concepts and definitions will then be modeled more formally in an ontology.

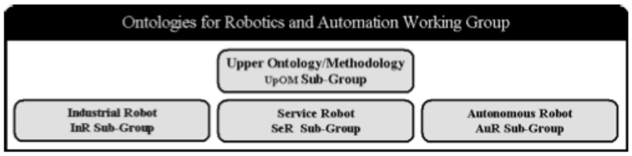


Figure 6: IEEE ORA Group Structure

The sub-domain ontologies will serve as a test case to validate the upper ontology and the methodology. The sub-domains were determined in such a way to ensure that there would be overlap amongst them. Once initial versions of the ontologies are completed, they will be integrated into the overall ontology. During the integration process, as overlapping concepts are identified, a process will be formalized to accurately 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.

For this effort, the working group has decided to use OWL (Web Ontology Language) [11] as the knowledge representation language. OWL is a family of knowledge representation languages for authoring ontologies and is endorsed by the World Wide Web Consortium (W3C). It is characterized by formal semantics and RDF/XML-based serialization for the Semantic Web. OWL was chosen by the group because of its popularity among the ontology development community, its endorsement by the W3C, as well as the number of OWL tools and reasoning engines that are available.

4. RELATED WORK

Performance evaluation of symbolic knowledge representation is not a new area; research has been explored for many years. Most of these research efforts have focused on the application of symbolic representations (specifically, ontologies) to domains that do not require real-time access and have primarily focused on the structure and consistency of the ontology as opposed to how it is applied to the domain.

In [12], Bhattacharya and Ghosh describe a generalized method for comparatively evaluating different knowledge representation schemes. They use expressiveness and performance as the primary metrics. Expressiveness is defined as the capability to correctly express the information appearing in one scheme in terms of the other scheme. Performance is defined as how resource "hungry" a knowledge representation scheme is with respect to processing, memory consumption, errors involved, etc. They evaluate systems based on criteria such as time complexity, space complexity, accuracy, relational capacity, maintainability, and user friendliness. As test examples, they use these metrics to perform pair-wise comparisons of rule-based schemes, object-oriented schemes, relational schemes, and hybrid schemes. They determined that hybrid schemes are best for the representation of zonation of landslide hazards, which is the domain they used for their study.

In [13], Aruna et. al. propose an evaluation framework made up of a number of different existing tools including OntoAnalyser [14], OntoGenerator, OntoClean [15], ONE-T, and S-OntoEval [16]. The supposition is that all of these tools provide different functionalities and benefits and that a combination of all of them is needed to perform a thorough ontology evaluation. The criteria that are proposed for evaluation include:

- Ontology properties
    - language conformity (syntax)
    - consistency (semantics)
- Technology properties
    - interoperability
    - turn around ability
    - performance
    - memory allocation
    - scalability
    - integration into frameworks
    - connectors and interfaces

The paper explains why this is important, but never goes into detail about how these tools can be combined into a common framework. It simply describes each tool without any conclusions.

In [17], Brank, Grobelnik, and Mladenic perform a survey of various ontology evaluation techniques. They describe evaluation approaches at various "levels," including lexical/vocabulary/data layer, hierarchy/taxonomy (and other semantic relationships), context/application level, syntactic level, and structure/architecture/design. They also describe various evaluation approaches and classify them as (1) comparing to a golden standard, (2) using ontologies in specific applications, (3) comparing ontologies with source data (e.g., collection of documents), and (4) evaluations performed by humans. They do not give opinions on which is best or worst… they simply try to classify the different approaches.

In [18], Gruninger and Fox describe the concept of competency questions to help evaluate ontologies. They start by defining scenarios that are relevant to the domain for which the ontology is being developed, and then develop competency questions that capture the questions that the ontology is intended to be able to answer. From these questions, concepts are identified and defined. There should be a direct mapping from the competency questions and the concepts, such that all of the concepts are present that allow the competency questions to be answered and no concepts are present that do not contribute to the answer to the questions. This approach focuses more on evaluating the concepts that are represented in the ontology as opposed other metrics such as performance related issues.

In [1], Vrandecic presents a theoretical framework and several methods for ontology evaluation with a focus on the Semantic Web. He focuses on the following three scenarios as relevant for ontology evaluation:

- Mistakes and omissions in ontologies can lead to the inability of applications to achieve the full potential of exchanged data. Good ontologies lead directly to a higher degree of reuse of data and a better cooperation over the boundaries of applications and domains.
- People constructing an ontology need a way to evaluate their results and possibly to guide the construction process and any refinement steps. This will make the ontology engineers feel more confident about their results, and thus encourage them to share their results with the community and reuse the work of others for their own purposes.
- Local changes in ontology development and maintenance processes may affect the work of others who are using the ontology. Ontology evaluation technologies allow a system to automatically check if constraints and requirements are fulfilled, in order to automatically reveal usability and compatability problems.

5. EXISTING METRICS FOR EVALUATING ONTOLOGIES

There are many different aspects of ontologies that one can analyze and measure. There are at least five significant additional research efforts that have attempted to capture some of these metrics. An excellent overview of ontology evaluation efforts is described in [1] and many of the descriptions below are adapted from this work. A superset of all of these metrics are listed below in alphabetical order, with pointers to the publications from which they arose. Some liberty was taken and assumptions applied to cluster metrics when significant overlap was perceived.

- Clarity/Understandability: The ontology should effectively communicate the intended meaning of defined terms. Definitions should be objective. When a definition can be stated in logical axioms, it should be. Where possible, a definition is preferred over a description. All entities should be documented with natural language. [19] [20] [21]
- Competency: The goals and purpose of the ontology is described using competency questions and the ontology has the concepts (and only the concepts) necessary to successfully answer the questions. [18]
- Completeness/Coverage: All the knowledge that is expected to be in the ontology is either explicitly stated or can be inferred from the ontology. [2] [20]
- Computational Integrity and Efficiency: the principle characteristics of an ontology that can be successfully/easily processed by a reasoner (inference engine, classifier, etc.). These could include logical consistency, disjointness ratio, etc, [21]
- Conciseness / Minimal Ontological Commitment: The ontology should specify the weakest theory (i.e., allowing the most models) and defining only those terms that are essential to the communication of knowledge consistent with that theory. [2] [19]
- Consistency/Coherence: capturing both the logical consistency (i.e., no contradictions can be inferred) and the consistency between the formal and the informal descriptions (i.e., the comments and the formal descriptions match) [2] [19] [20]
- Expandability/Extendibility: An ontology should offer a conceptual foundation for a range of anticipated tasks, and the representation should be crafted so that one can extend and specialize the ontology monotonically. New terms can be introduced without the need to revise existing axioms. [2] [19]
- Mappability to upper level and other ontologies [20]
- Minimal encoding bias: An encoding bias results when representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized, because knowledge-sharing agents may be implemented with different libraries and representation styles. [19]
- Relevance: Evaluation against specific use cases, scenarios, requirements, applications, end-user knowledge, and data sources the ontology was developed to address [20]
- Reusability/Flexibility: How easily the developed ontologies can be applied to unanticipated domains that require the same sort of knowledge or lend itself to various views. [20] [21]
- Sensitivity: relates to how small changes in an axiom alter the semantics of the ontology. [2]
- Soundness: Free from error [20] [21]
- Types of inferences that can be used [20]
- Usability/Organization Fitness: Compliance to procedures for extension, integration, adaptation, and access for effective application. Can it be easily deployed within an organization? [21]

This information in tabular form is included below:

Table 1: Ontology Evaluation Metrics

Metric	Gangemi [21]	Gomez -Perez [2]	Gruber [19]	Gruninger [18]	Obrst [20]
Clarity / Understandable	x		x		x
Competency				x	
Completeness / Coverage		x			x
Computational Integrity and Efficiency	x				
Conciseness / Minimal Ontological Commitment		x	x		
Consistency / Coherence		x	x		x
Expandability / Extendability		x	x		
Mappability					x
Minimal Encoding Bias			x		
Relevance					x
Reusability / Flexibility	x				x
Sensitivity		x			
Soundness	x				x
Types of Inferencing					x
Usability / Organization Fitness	x				

It is interesting to note the relatively minimal overlaps between the metrics mentioned in each of the papers. There is no metric that shows up on more than three of the research papers and this only happens two times. In addition, eight of the metrics only show up once in the five research papers. This could be due to a number of factors:

1. There is not broad agreement in the community about the metrics that should be used to evaluate ontologies.
2. There is some overlap among the requirements such that the same things are evaluated but are categorized differently. This could be due to the liberties that were taken by this paper's author to categorize the metric descriptions in the respective papers or from different sets of terminologies used by each paper's author.
3. The authors focused on specific aspects of ontology evaluation and did not try to take a comprehensive view of all of the aspects involved.

It is likely that this lack of overlap is due to some combination of all three items above, though it is the authors' belief that item #1 (lack of broad agreement) is the most substantial.

6. WHERE ARE THE GAPS?

Robots are innately real-time systems. However, real-time is a relative word. At the servo level, real-time can mean tens or hundreds of cycles per second. At the higher-level planning level, real-time can be on the order of tens of seconds or minutes (or even longer). The trick is to figure out where symbolic representations like ontologies play a role, both in the usefulness of the information that they provide and in the representation's ability to work within a system to deliver information at the rate necessary.

Many of the lower-level real-time aspects have been removed from the symbolic representation realm and applied to other types of representations that are better suited for them (e.g., parametric and spatial knowledge levels, as discussed in Section 2). While this has worked in the past, symbolic representations provide a level of information that would be valuable to real-time applications, including the ability to reason over existing knowledge at a level deeper than what is possible in other types of representations. As can be seen in Section 5, almost all of the metrics focus on the structure of the ontology, including clarity, completeness, relevance, sensitivity, soundness, etc. Almost none of the metrics focused on the functionality that the ontology supports, such as how quickly it is able to work within a system to process new data or how rapidly it is able to work within a system to provide useful data back to the application. This is alluded to in the metric "computational integrity and efficiency" but this was just presented as a concept in the literature without details of how one would go about analyzing it and how one would determine if the resulting metrics are suitable for the application of interest.

One area that will be explored in the future is coupling the ontology with other types of symbolic representations, such as databases, that may be able to handle real-time applications more efficiently at lower levels in the control hierarchy. In concept, there are several data structures in the ontology which would not need to be updated in real time and would likely stay static throughout an entire ontology application. This may include the names of certain objects, their capabilities, and in the case of static items, their locations. For example, in a manufacturing plant performing automated kitting operations, the names of the machines, their locations, and their capabilities may stay the same during the entire operation. However, the exact location of their robotic arm, what kit they are working on at the time, and the parts that are being manipulated may change by the minute or second. The idea is that these "dynamic" concepts would have a link from their instances and structures in the ontologies to a database that would be dynamically updated as new information is made available from the sensor systems (or entered by a human).

Information can either be "pushed" from the database to the ontology instances when some criterion is reached (e.g., an object's location is moved by over a predefined distance, the state of the overall system reaches a milestone, an error state is detected, etc.), or can be "pulled" from the database to the ontology at certain time intervals or just before reasoning is about to be performed. With this approach, a system would rely on the database structures for the real-time access and updating functions but would still get the benefit of ontology reasoning through the links between the database and the ontology.

Another advantage of this approach is the reusability and semantics that the ontology provides that may not be available through the database alone. Databases are very good at representing concepts and their characteristics, but do not provide detailed semantics about what the concepts and characteristics mean. By coupling the database fields with the ontology instances, detailed semantics can be captured in the ontology while not slowing down the processing of the information in the database.

Once the application is concluded (e.g., a kitting operation), the resulting database information can be written back to the ontology and easily shared with other applications. This could include scheduling systems, process planning systems, or other management-type applications that have a need to see and understand the state of the factory at any given time. Ontologies are often developed to be highly reusable, thus providing another benefit of the database-ontology integration.

7. CONCLUSION

In this paper, we discuss some of the ways that knowledge is represented in robotic applications, describe an IEEE effort to standardize symbolic representation in robot systems, look at some metrics that have been applied to measuring the quality of symbolic representations, and provide thoughts on what other types of metrics and procedures may be necessary to measure the performance of symbolic representations (with an emphasis on ontologies) in robotic applications. This is the first paper in what is expected to be a series of papers detailing ways to measure and apply symbolic representations to the robotics field. With much of the research in this area not yet started, the purpose of this paper is to describe some related efforts and some preliminary thoughts that will set the stage for future work.

DISCLAIMER

The name of commercial products or vendors does not imply NIST endorsement or that this product is necessarily the best for the purpose.

REFERENCES

[1] D. Vrandecic, "Ontology Evaluation," PhD, Institute of Applied Informatics and Formal Description Methods, Karlsruher Instituts fur Technologie (KIT), 2010.

[2] A. Gomez-Perez, "Ontology evaluation," in *Handbook on Ontologies*. vol. First Edition, S. Staab and R. Studer, Eds., ed: Springer, 2004, pp. 251-274.

[3] E. Messina, J. Albus, C. Schlenoff, and J. Evans, "Knowledge Engineering for Real Time Intelligent Control," *Journal of Intelligent & Fuzzy Systems,* vol. 14, 2003.

[4] "ISO 10303-21, Industrial automation systems and integration - Product data representation and exchange - Part 21: Clear Text Encoding of the Exchange Structure," Geneva, Switzerland1994.

[5] T. Gruber, "A Translation Approach to Portable Ontology Specification," *Knowledge Acquisition,* vol. 5, p. 22, 1993.

[6] D. Lenat, R. Guha, K. Pittman, D. Pratt, and M. Shephard, "CYC: Toward Programs with Common Sense," *Communications of the ACM,* vol. 33, pp. 30-49, 1990.

[7] M. Genesereth and R. Fikes, "Knowledge Interchange Format," Stanford University.

[8] W. F. Clocksin and C. S. Mellish, *Programming in Prolog*. New York: Springer-Verlag, 2003.

[9] R. Madhavan, Yu. W, Biggs, G., Schlenoff, C., Huang, H.M., "IEEE RAS Standing Committee for Standards Activities: History and Status Update," *Journal of the Robotics Society of Japan, Special Issue on "Activities of International Standards for Robot Technologies"*, vol. 29, May 2011.

[10] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelanez, "An IEEE Standard Ontology for Robotics and Automation," in *to be published in Bridges Between the Methodological and Practical Work of the Robotics and Cognitive Systems Communities - From Sensors to Concepts*, A. Chibani, Ed., ed: Springer-Verlag, 2012.

[11] (2009). *W3C OWL Working Group, OWL 2 Web Ontology Language Document Overview*. Available: http://www.w3.org/TR/owl2-overview/

[12] D. Bhattacharya and J. K. Ghosh, "Evaluation of Knowledge Representation Schemes as a Prerequisite towards Development of a Knowledge-Based System," *Journal of Computing in Civil Engineering,* November/December 2008 2008.

[13] T. Aruna, K. Saranya, and C. Bhandari, "A Survey of Ontology Evaluation Tools," presented at the International Conference on Process Automation, Control and Computing (PACC), Coimbatore, India, 2011.

[14] D. Rogozan and G. Paquette, "Managing Ontology Changes on the Semantic Web," presented at the IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), France, 2005.

[15] A. Oltramari, A. Gangemi, N. Guarino, and C. Masolo, "Restructuring WordNet's Top=Level: The OntoClean approach.," presented at the LREC2002 (OntoLex workshop), Las Palmas, Spain, 2002.

[16] R. Dividino, M. Romanelli, and D. Sonntag, "Semiotic-based Ontology Evaluation Tool (S-OntoEval)," presented at the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, 2008.

[17] J. Brank, M. Grobelnik, and D. Mladenic, "A survey of ontology evaluation techniques," presented at the Conference on Data Mining and Data Warehouses (SikDD), Ljubljana, Slovenia, 2005.

[18] M. Gruninger and M. Fox, "Methodology for the Design and Evaluation of Ontologies," presented at the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal Quebec, 1995.

[19] T. Gruber, "Towards principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies,* vol. 43, 1995.

[20] L. Obrst, W. Ceusters, I. Mani, S. Ray, and B. Smith, "The evaluation of ontologies," in *Revolutionizing Knowledge Discovery in the Life Sciences*, C. J. O. Baker and K.-H. Cheung, Eds., ed: Springer, 2007, pp. 139-158.

[21] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann, "Ontology evaluation and validation: an integrated formal model for the quality diagnostic task," Rome, Italy, 2005.