Proceedings of the ASME 2017 International Manufacturing Science and Eng. Conference
MSEC2017
June 4–8, 2017, Los Angeles, CA, USA

MSEC2017-2783

TOWARDS A ROBOT TASK ONTOLOGY STANDARD

Stephen Balakirsky *
Georgia Tech Research Institute
Atlanta, USA
Email: steveb@gatech.edu

Craig Schlenoff
National Institute of Standards and Tech
Gaithersburg, USA

Sandro Rama Fiorini
Lissi - Université Paris-Est Créteil
Vitry-sur-seine, France

Signe Redfield
Robotistry.org
Pomfret, USA

Marcos Barreto
Federal University of Bahia (UFBA)
Salvador, Brazil

Hirenkumar Nakawala
Politecnico di Milano
Milan, Italy

Joel Luís Carbonera †
IBM Research
Rio de Janeiro, Brazil

Larisa Soldatova
Brunel University
London, UK

Julita Bermejo-Alonso
Univ. Politécnica de Madrid
Madrid, Spain

Fatima Maikore
Brunel University
London, UK

Paulo J.S. Goncalves
IDMEC, IST, Instituto Politecnico de Castelo Branco, Portugal

Elena De Momi
Politecnico di Milano
Milan, Italy

Veera R. Sampath Kumar
Monash University
Malaysia Campus, Malaysia

Tamás Haidegger
Óbuda University, IROB Center
Budapest, Hungary

ABSTRACT

Ontologies serve robotics in many ways, particularly in describing and driving autonomous functions. These functions are built around robot tasks. In this paper, we introduce the IEEE Robot Task Representation Study Group, including its work plan, initial development efforts, and proposed use cases. This effort aims to develop a standard that provides a comprehensive ontology encompassing robot task structures and reasoning across robotic domains, addressing both the relationships between tasks and platforms and the relationships between tasks and users. Its goal is to develop a knowledge representation that addresses task structure, with decomposition into subclasses, categories, and/or relations. It includes attributes, both common across tasks and specific to particular tasks and task types.

INTRODUCTION

Industrial automation is steadily evolving towards computer-controlled processes over fixed automation. For example, manufacturing environments ranging from small batch to large scale production desire seamlessly integrated processes and systems. All actors need to have sufficient knowledge of their tasks to not only perform them, but to also communicate their pending activities to others and to recognize and correct errors without the need to interrupt the process. In this context, *task* refers to a concrete decomposition from goal to subgoals that enables

*Address all correspondence to this author.
†Previous affiliation: UFRGS, Porto Alegre, Brazil

the human/robot to accomplish the goal at a specific instance in time. The tasks can be either informational (i.e., storing, representing, or transferring information between the actors) or physical, where the actors actually manipulate materials (e.g., the robot picks and places an object) [14]. The task could also be " collaborative" (e.g., human–robot manipulation in materials handling), where autonomous robots are expected to collaborate with other robots, as well as humans. With growing demand for product variations, limited skilled manpower, increased complexity of working environments, and greater requirements for robot–robot and human–robot collaboration, there is a need for a standard providing an explicit knowledge representation for robot tasks. The availability of such a standard knowledge representation will:

- define the domain concepts as controlled vocabularies for robot task representation;
- ensure a common understanding between different industrial groups and devices;
- facilitate interoperability between robotic systems for efficient data transfer and information exchange;
- increase manufacturing performance (e.g., flexibility) by more easily incorporating new processes due to a common set of concepts.

The purpose of this paper is to introduce the *IEEE Robot Task Representation* (RTR) Study Group, whose goal is to develop a broad standard that provides a comprehensive ontology for robot task structure and reasoning with specific application to the industrial robotics domain and its subfields. This paper describes the procedures RTR will follow to accomplish its goal, and describes the initial work performed to define terminology serving as the basis for the ontology. This work will be a supplement to the existing standard Core Ontology for Robotics and Automation (CORA) [1], which was developed by the IEEE Ontologies for Robotics and Automation Working Group. This supplement will include the presentation of concepts in an initial set of application domains (e.g., manufacturing) where robot task representation could be useful. The ontology provides a unified way to represent knowledge about robot tasks by sharing common knowledge and preserving meaning. It can be utilized in manufacturing control applications, where the system needs to control multiple elements of the manufacturing process and information needs to be shared among them.

From the robot task perspective, an ontology provides a knowledge representation of key concepts related to robot tasks with properties, relationships, and constraints relevant to the industrial processes. An ontology provides a richer set of relationships between domain concepts.

Our work plan for developing the standard has two aspects. Firstly, to develop the task ontology, extending CORA and capturing vocabularies for robot task representation by requirements analysis and surveying the literature. The final decision-making on vocabularies will be achieved through consensus between different group members. The ontology will contain vocabularies for generic tasks and specialized tasks for the industrial application. The second aspect is to develop a task repository, which will provide a set of instances that could be used for robotic implementation and validation of the task ontology.

The rest of the paper follows the following outline: the *Related Work* section discusses some recent efforts for developing an ontology for robot task representation, the *SUMO/CORA* section represents background on work performed for the IEEE 1872-2015 standard, the *Work Plan* section discusses the structure of the effort, requirements gathering, and the overall development of the ontology, and the *Terminology* section represents a first attempt to build terminology for robot tasks based on foundational ontologies and CORA. In the *Use Cases* section, we discuss use cases representing potential applications in various domains, and conceptual similarities between the use cases, and the *Summary and Next Steps* section concludes the paper.

RELATED WORK

The set of tasks a robot is expected to perform depends on the application domain (e.g., industrial, autonomous, service, surgical) it belongs to. There are standard tasks that apply across multiple domains and, within each area, application-specific ones. This mixture between standard and specialized tasks poses a challenge related to how to efficiently and unambiguously represent them.

Many in the literature have defined a task as an action with proper input and output information that moves the application from one state to another. In robotics, a set of tasks can be expressed in different ways, including domain definition languages, finite state machines (FSM), and ontologies. Domain definition languages present a good coverage of objects, actions and behavior from the domain they represent, but tend to be very specific or limited to this domain. In general, they are more targeted to planning actions, expressing the domain in terms of predicates and actions and the problem to be solved as objects, initial state and goal state [13].

FSM favors the knowledge a robot must acquire to perform the set of tasks and eases the communication among robots or between humans and robots. However, it has limited power to express some specific requirements such as temporal requisites or intra-task dependencies. There are a few examples in the literature to consider: a Mealy state machine is used in [12] to represent assistive actions a robot can perform while observing a human performing a table assembly. In this context, a task is represented by a set of states, and the robot is expected to identify and react to each by performing HOLD and ROTATE actions. A speech system engine based on OpenHRI is also used to allow the robot to generate verbal messages to explain actions and describe state to the user. A behavior-based model [9] rep-

resents complex tasks involving temporal constraints, intra-tasks dependencies, and multiple paths a robot can use to finalize a running task. Robots are organized in "goal" and "behavior" nodes that follow constraints such as THEN, AND, OR, and WHILE to perform reasoning, generate messages, and assess task state and completion.

Machine learning can also be used to address the complexity related to task representation, reasoning (knowledge), and communication among robots. In [18], robots improve their knowledge about a motor skill learning scenario and evaluate a set of learning algorithms as they allow for dynamic changes in their environment. Tasks are represented through a small set of primitive motions a human performs while moving. Motor skills are acquired through a set of supervised and reinforcement learning steps.

Within the robotic domain, there are several attempts to formalize the task knowledge robots use in different applications: Proteus is an ontology that describes mobile robotics scenarios as a set of modules where the robotic elements and the missions are defined [15]; KnowRob is part of an ontology-based robot knowledge framework for service robots [22]; Ontology for Robotic Orthopedic Surgury (OROSU) is an ontology for robotic surgery applied in the biomedical field [11]; RoboEarth defines an ontology to facilitate task knowledge exchange among heterogeneous robots [23]; and Ambient Assisted Living (AAL) ontologies used in cloud robotics [24]. As for the manufacturing domain, ontologies have been used in flexible manufacturing systems [25] and manufacturing design of new products [4].

When robots and manufacturing are combined, an ontology-based approach allows the robot to translate new orders within the manufacturing process into what tasks to perform and how to perform them. This kind of ontology for agile manufacturing is described in [3]. It provides a knowledge base for static and dynamic information about parts and actions supporting failure analysis in robotic cells.

Given these different domains and the mixture of standard and specialized tasks, a task ontology provides an efficient way to describe the vocabulary for a generic task or activity, specializing the terms of the top-level (or domain) ontology. Such an ontology should capture task related knowledge from both structural and behavioral perspectives as represented by the domain knowledge (objects, relations, events as knowledge roles to be played by domain concepts while performing a task) and the problem-solving knowledge (how to achieve the defined goals, generally as task decomposition into subtasks and control flow) [5, 10, 16]. The proposed extension to our CORA ontology is an approach to address both perspectives.

CORA

The IEEE 1872-2015 standard [1] defines a set of ontologies related to Robotics and Automation (R&A), chiefly the Core Ontology for Robotics and Automation (CORA). Core ontologies capture the main concepts and relationships in a given domain.

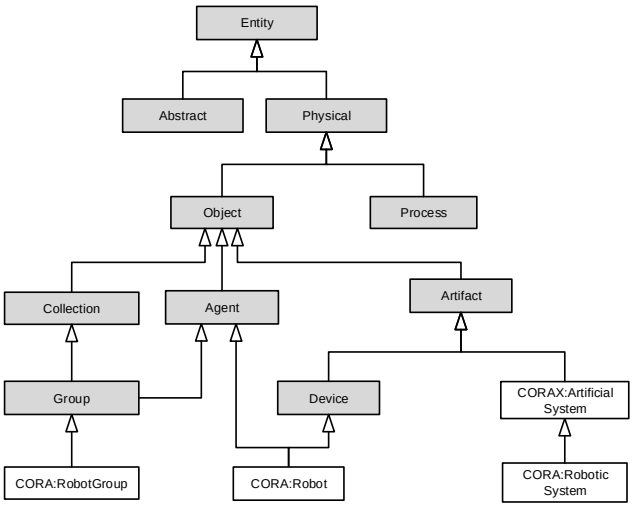

FIGURE 1. Overview of the main concepts in SUMO (shadowed) and CORA

There are two approaches by which CORA can be extended and put to use. First, various groups and institutions can provide specialized ontologies that extend CORA into more specific domains of R&A. This is currently taking place in the sub-domains of autonomous robots [17] and medical robotics. The other approach is to complement CORA with additional high-level ontologies. CORA addresses only a subset of the general concepts within R&A. This paper describes one of these complementary concepts not addressed beforehand – in this case, the notion of robot task.

SUMO/CORA

We must extend task representation concepts in CORA in order to define the Robot Task Ontology in conformance with IEEE 1872-2015. However, CORA itself extends the Standard Upper Merged Ontology (SUMO) top ontology, which provides CORA with very high level ontological distinctions about reality. Thus, extending CORA means extending SUMO/CORA. For that reason, we provide a brief overview of both ontologies [1, 19]).

As shown in Fig. 1, SUMO divides all entities that exist into two large groups: physical and abstract. Physical entities have "*a location in space–time*". Abstract entities do not exist in space-time (e.g., mathematical and epistemological entities). Physical entities are separated into *objects* and *processes*. *Object* corresponds to the class of ordinary objects, including spatial regions. *Processes* on the other hand are entities that have temporal parts and that are not objects.

CORA describes what a robot is by extending concepts in SUMO. It defines three main entities related to robot: *robot*,

robot group, and *robotic system*. The term *robot* may have many definitions, and CORA acknowledges this inherent ambiguity as an intrinsic feature of the domain, and therefore defines robot purely based on necessary conditions. CORA uses a particularly broad definition of robot, at the cost of including what some roboticists may think of as non-robot entities. CORA states that a *robot* is a *device* in the sense of SUMO. According to SUMO, a device is an artifact (i.e., a *physical object product of making*), which participates as a tool in a process. Being a device, robot inherits the notion that devices have parts from SUMO, allowing the representation of complex robots with robot parts. A robot is also an *agent*. SUMO states that agent is "*something or someone that can act on its own and produce changes in the world*". Robots perform tasks by acting on the environment or themselves and they can form robot groups. A *robot group* is also an agent in the sense that its own agency emerges from its participants. This notion can be used to describe robot teams, or even complex robots formed by many independent robotic agents acting in unison. *Robotic systems* are systems composed of robots (or robot groups) and other devices that facilitate the operations of robots. A good example of a robotic system is a car assembly cell at a manufacturing site. Actuated structures within the environment manipulate the car body enabling the industrial robots within the system to act on it. An environment equipped with a robotic system is a *robotic environment*.

WORK PLAN AND DEVELOPMENT APPROACH

The goal of the RTR Study Group is to develop a knowledge representation that addresses task structure, with decomposition into subclasses, categories, and/or relations. There are two parts of the proposed standard: the *Task Ontology* and the *Task Repository*.

Although the Task Ontology will be the official standard when completed, the Task Repository is necessary to help validate the standard, and to provide an avenue that makes the standard more useful and applied in the industry. The Task Ontology formally defines what a task is, and specifies the properties of tasks, the properties of the hierarchy in which tasks are placed, and the ways in which the performance of the capabilities required to accomplish the tasks are measured. The Task Repository enables the community to build up a shared catalog of tasks and capabilities along with their relationships (based on elements within the Task Ontology). The purpose of the overall standard is to ensure common representations and frameworks when tasks are described, so the knowledge represented in the Task Ontology defines the structure and content of the tasks in the Task Repository. The purpose of this representation is to enable:

(i) communication of task-related data among different stakeholders;
(ii) categorization of tasks according to a variety of criteria;

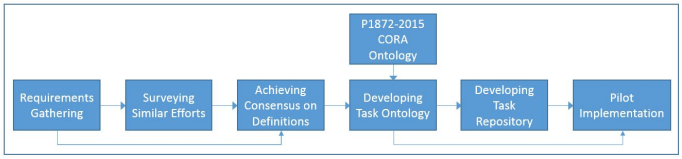

FIGURE 2. Robot Task Representation Study Group Work Plan

(iii) improved awareness and identification of potentially common solutions to disparate problems;
(iv) support for alignment of existing capabilities to support of new tasks;
(v) support for alignment of new capabilities to support existing tasks.

The existence of such a repository allows for clear definitions and descriptions of tasks and the ability for a user to quickly and easily determine if a task description (and associated algorithm) exists that will accomplish their goal, even if that task may have been created for a different purpose in another domain.

To develop this ontology, we will perform the following:

1. **Requirements Gathering:** It is important to gather the relevant information requirements from all of the target domains, to ensure that the resulting knowledge representation is truly comprehensive. RTR will reach out to experts in various robotics fields to gain a deep understanding of what is necessary to represent task information in their domains. In addition to compiling the terms and definitions, RTR will start the process of identifying cases in which the same term has different meanings in different domains, as well as the cases where the same definition is associated with different terms. The output of this process will be a glossary of terms and definitions, sorted by robot domain, which will serve as the basis for subsequent steps in the work plan.
2. **Surveying Similar Efforts:** As discussed in the Relevant Work section, there are a number of efforts that have attempted to capture task information in specific robotic domains. In this phase of the work plan, RTR will deeply analyze these efforts to see how well they capture the concepts identified in the step above and evaluate their potential use as sources of definitions for the main concepts identified. In addition, RTR will look at the way that these concepts are represented (i.e., the knowledge representation formalism that is used and the attributes and relationships that are represented) to leverage the existing representational approaches and concepts wherever possible.
3. **Achieving Consensus on Definitions of Terms:** Coming to consensus on terms and definition is the most challenging, yet essential part of the work. We have formed the study group so that there is representation from a wide array of robot domains. We are also closely collaborating with the autonomous robotics subarea, which is also extending the

1872-2015 standard.

4. **Developing the Task Ontology:** Building on the previous work of 1872-2015 (CORA), and using consensus definitions, RTR will extend the CORA ontology to capture terms and definitions pertaining to robot tasks. While CORA does not specifically address robot tasks, it does define high level concepts of robot motion that can be leveraged. In addition, SUMO defines the concept of process, which can also be built upon as appropriate. RTR will use aspects of the METHONTOLOGY ontology development methodology [8], which includes the development of the ontology conceptual model; the formalization of the ontology, which involves transforming the conceptual model into a formal or semi-computable model (specified in first-order logic, for example); and the implementation of the ontology, which involves transforming the previously formalized ontology into a computable model, codified in an ontology representation language, such as the Web Ontology Language (OWL).
5. **Modeling of Shared Tasks and Capabilities in a Task Repository:** While the Task Ontology provides the structure and definitions of concepts, the Task Repository provides task instances that can be applied towards robot applications. It uses the structure of the concepts in the Task Ontology and populates the values with instances specific to individual robot types and applications. To validate the Task Ontology, RTR will create a set of task instances in the Task Repository, focusing on tasks that are generally applicable across robot domains, such as pick and place tasks or robot mobility tasks. The hope is that as the Task Ontology and associated Task Repository get used, the community will populate the Task Repository with additional task instances.
6. **Pilot Implementation:** Using the task instances created in the previous step, we will create two control systems in different domains. These control systems will use the knowledge represented in the task repository to control a robot performing operations in the domain of interest. We expect one of the domains to be in the manufacturing field due to the interest and availability of these types of robots among the study group members. The second domain is still to be determined. As mentioned above, we expect the tasks to focus on pick and place and mobility operations. This process will help to validate the knowledge represented in the Task Repository as well as the structure of the Task Ontology.

TERMINOLOGY

This initial set of terminology (and associated definitions) are the basis for discussion. While RTR has not yet reached consensus on the definitions, it presents the types of terms being addressed, and the initial concepts on their definitions. These terms are broken into four categories, related to task descriptions, properties, implementation aspects, and context. In each section, a small set of terms is included with definitions that address specific contexts and usages, where lower case roman numerals identify each definition. When a specific term's definition is referred to in the subsequent text, it is indicated by providing a subscript after the term corresponding to its definition (e.g., task_i).

Terms and Definitions

Task Description Terminology Task description terminology defines what the user or operator needs to get done.

Goal—What the robot is attempting to accomplish.

i The externally defined desired end (or continuing) state of the system. Note that the goal_i is the action that the operator or other external entity wants the robot to do. If the task is decomposed to subtasks, the goal_i is the desired end (or continuing) state of each subtasks, as defined by the task.

ii A subsidiary desire within the context of a larger problem—goal_{ii} is shorthand for what is trying to be accomplished in the abstract (e.g., as the operator or designer would define it). If the overall goal_i is to move a part to a particular location, the robot or its operator may be told to have a goal_{ii} of reaching a specific interim pose.

iii *Colloquial*: A metric against which a given performance is evaluated in the context of a specific task_i (applies when goal_i or goal_{ii} involve quantitative elements, e.g., "The goal_{iii} is to move 10 parts.")

Task and Subtasks—The concrete decomposition from goal_i to sub-goals_{ii} that enable the robot to accomplish the user's goals_i. This term is particularly problematic, since practitioners in different fields tend to define it differently, and even within any given field, its usage is arbitrary.

i A restatement of the goal from the robot's perspective. If the goal_i is the expression of what the operator wants done, the task_i is how the robot interprets it (i.e., task planning). Subtasks can be defined to whatever depth it is necessary. Tasks and subtasks are accomplished via behaviors and actions.

ii A lower or higher level behavior. Within a given discussion, it is common for task_{ii} to be used as a generic term to enable individuals to differentiate between the behavior or action under discussion and other lower or higher level actions. This is particularly relevant during task_i decomposition/task_i allocation discussions, where the decomposition process results in sub-tasks that, from the perspective of the original task_i, are synonymous with the actions used to accomplish them.

Note that there are as many ways of breaking up a goal_i into tasks_i and subtasks_i as there are robots and autonomy designers. Furthermore, there is considerable confusion regarding the exact definition of "task" and "behavior", as they are commonly used.

"Task" is often used to describe both the $goal_i$ and the behavior designed to accomplish it, and the words used to define a specific $task_i$ are often the same words used to describe behaviors. There are some attempts to separate a generic behavior programmed into the robot ("pick up a part") from a specific behavior instantiated by the robot ("pick up that part"), but often no distinction is made ("place part" can refer to the $goal_i$, the $task_i$ and the behavior used to accomplish it).

Task Property Terminology Task property terminology defines properties of the job.

Constraint—Constraint properties limit the ways in which the robot can complete the task. They help to define the conditions under which the task will be considered as accomplished.

i A limitation defining the desired properties of the task that can be expressed as a functional requirement (examples: complete task within 10 seconds; stay inside the safe zone)

Evaluation—Evaluation properties are concerned with how the robot's performance of the task is evaluated. These include both metrics that are used to evaluate capabilities, and are largely independent of the task itself and metrics that are used to evaluate the performance of the task.

i A performance metric defining a desired property of an action or capability (e.g., path smoothness, shortest path length)
ii A performance metric defining acceptable performance of a given task (e.g, all parts were placed in the correct positions)

Task Implementation Terminology Task implementation terminology defines the aspects of the $task_i$ that relate to specific things that must be accomplished. While in some cases ("find my keys"), $tasks_i$ can be defined independently of the robot's actions, in others, specific actions are a key component of the $task_i$ description ("attach this bolt"). This section is concerned with the terminology we use to describe components of the $task_i$ related to robot activity.

Action—Actions include any physical motion or desired activity, physical or virtual.

i physical motion, desired or instantiated (example: move arm to pose)
ii low level behavior that may or may not result in physical motion (example: turn on sensor)
iii the lowest level behavior in the context of a given discussion or system (e.g., reach joint angle)

RobotBehavior is the CORA implementation of Action.

Capability—Capabilities are concerned with action in the abstract.

i a generic term referring to an $action_i$ or $action_{ii}$ the robot has been programmed to perform, a skill the robot provides, or a behavior the robot has available. Can be used at any level. (e.g., Action = close gripper or move arm; Skill = push, pull, or pick up part; Behavior = pick and place)

Task Context Terminology Task context terminology enables the user to define relevant properties of the context in which the task is expected to take place. This context takes two forms: the environment, which includes both the larger external context and the things that the robot is expected to interact with and the platform, which includes any properties the robot needs in order to accomplish the task.

Environment—things that are not part of the robot system

i the physical world in which the task is situated (e.g., two dimensional vs. three dimensional; underwater vs. ground vs. air)
ii the objects which the robot is expected to interact with or not interact with as part of accomplishing the task (e.g., the cup in "pick up cup"; a pedestrian in an autonomous driving situation)

Platform—things that are part of the robot itself

i the physical properties of the robot (e.g., wheels vs. treads; battery vs. fuel)
ii the functional properties of the robot (e.g., range sensor vs. color sensor; holonomic vs. non-holonomic)

In manufacturing, a robot arm may be required to assemble components. $Goal_i$ could be "part A and part B are rigidly connected at point C". $Tasks_{ii}$ would include "pick up part A" and "pick up part B", as well as "hold part A next to part B". "Move to pose X relative to part A" would be a $subtasks_{ii}$ of "pick up part A". We could impose the $constraint_i$ that the robot arm not move outside a certain portion of its workspace, and evaluate its performance based on how long it takes to complete the job ($evaluation_i$). It would require a "grasp part" $action_{ii}$ and "identify part A" and "identify part B" $capabilities_i$. Its $environment_i$ would be defined by the factory in which it was expected to work and its $environment_{ii}$ would also be defined by part A and part B, and the $platform_i$ would be an instance of a robot arm.

Tasks in SUMO/CORA

Neither SUMO nor CORA define a specific concept for task. The WordNet/SUMO alignment maps the different WordNet meanings of the term *task* to different subclasses of Intentional Process in SUMO. An Intentional Process is a *process that has a specific purpose for the agent who performs it*. Therefore, an instance of a task in SUMO is an entity that has a location in space-time, much like an event, and that has a purpose for a given agent. It approximates the sense of $task_{ii}$ in the previous

section. However, SUMO also restricts intentional processes to processes acted upon by cognitive agents, which are defined as biological organisms. Thus, Intentional Process cannot be used as a basis for our definition of Task in Robotics.

In relation to task decomposition, SUMO defines the concept Plan as *a specification of a sequence of processes which is intended to satisfy a specified purpose at some future time.* The concept is located in the abstract branch of SUMO, being a subclass of Proposition and Procedure (Plan $\rightarrow$ Procedure $\rightarrow$ Proposition $\rightarrow$ Abstract). While the provided axiomatization of Plan and Procedure is not rich enough to draw deeper conclusions about their meaning, it is interesting to note that Plan is defined as a kind of Proposition. A Proposition in SUMO is an abstract entity that represents a thought. For example, the sentence "the book is on the table" expresses the proposition that there is a book situated on top of a particular table. This sentence in e.g., Portuguese ("o livro está sobre a mesa") is a different sentence that express the same proposition. Propositions are materialized by instances of Content-bearing Object, which is a kind of physical object that represents one or more propositions, such as the two pieces of text above. SUMO separates information (the Proposition) from how it is physically represented or encoded (the Content Bearing Object). Content-bearing objects also include non-linguistic objects, such as pictures and icons. Furthermore, SUMO introduces the relationship *realization*, which allows one to link instances of propositions with the instances of processes that express them.

In summary, SUMO allows one to represent a situation where a plan (or a procedure), encoded in one or more documents (such as computer files) is realized by instances of processes at different times. We can go a step further by bringing in the notion of Robot Behavior from the *Robot Architecture Ontology* (RAO) [6]. The RAO ontology aims to extend CORA with concepts and relationships about robot architectures. It defines the concept Robot Behavior as any process that has a robot as agent. Thus, SUMO/CORA/RAO. A Plan is is only expressive enough to represent robot behaviors and their relation to hypothetical plans they might act upon.

Initial Ontology Specification

Because the Task Ontology extends SUMO and CORA as described in previous sections, the structure of the Task Ontology is aligned with their structures. We take inspiration from SUMO, defining Plan and separating our task-related terminology within three main branches:

- *Object* branch, e.g., *Robot*, *Platform*;
- *Process* branch, e.g., *Robot Behavior*;
- *Proposition* branch, e.g., *Task*, *Goal*.

Fig. 3 shows how the key terms connect to each other. A robot can have the capability to solve various tasks. This is expressed via the relation *solves* between *Robot* and *Task*, and the relation *hasCapability* between *Robot* and *Capability*. Goals are at a higher level, and connected to tasks at a lower level via the relation *abstractPart*, where *Task* is realized via *RobotBehavior*. A *Task* depends on both *Environment* and *Platform*. *Tasks* usually carry *Constraints* and may have *Subtasks*.

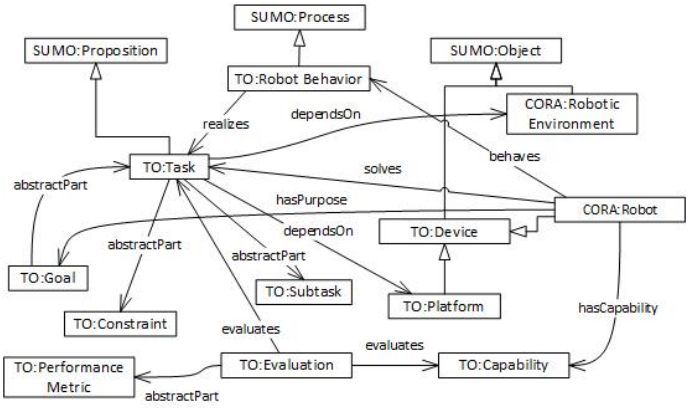

FIGURE 3. The structure of the Task Ontology (TO) (a fragment); solid lines represent sub-class relations and dashed lines correspond to other relations

.

USE CASES

In order to be truly useful, a standard must be applicable across a large section of the industry that it is aimed to serve. In the case of a task ontology for robotic systems, domains ranging from industrial automation to self-driving cars and research robots could potentially benefit from a standardized representation. The mission for this working group is to determine where there are similarities between the structure of the tasks in the various domains and how to coherently represent these similarities across domains. In order to perform this mission, RTR will develop domain specific task models by examining robot tasking in diverse domains and situations. We anticipate that significant areas of overlap will be discovered when we compare the models for the various domains. This standard will concentrate on these areas of overlap. The use cases presented below, derived from the authors' current work, are being examined to develop these models. In the text below, we showed which definitions of specific terms are being used (as indicated by the subscript after terms such as task and goal) to indicate where overlap exists and where the same term is being used with different meanings.

Agile Manufacturing

Many of today's assembly lines perform mixed-model assembly, where the same line produces many slightly different variants of the same product. According to Zhu, et al. [27] there

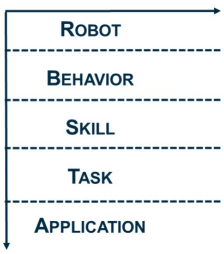

FIGURE 4. Task structure utilized by agile manufacturing use case.

are 10^{17} possible variations of the BMW 7 Series. Robotic automation should support automatic reconfiguration to construct any of these variants without the need for re-programming. In addition, many robot vendors make products that have similar capabilities and could be utilized interchangeably in applications, but these robots need to be programmed in custom languages, and small- and mid-sized manufacturers do not have the resources to learn to program more than one variant. These problems lend themselves to a hierarchical and modular decompositions of the top-level robot application. Figure 4 shows one such decomposition from Balakirsky et.al [2].

Proctor et al. [20] have taken the first steps in developing a vendor independent robot programming language at the lowest (robot) level of decomposition. This hierarchical language (known as the Canonical Robot Command Language or CRCL) allows the robot to be given a task$_{ii}$ consisting of movements in Cartesian space, and works identically on multiple robots from multiple vendors. The next higher level defines behaviors which constitute the open-loop application of sequences of CRCL commands. An example of a behavior would be following a path that is known to be obstacle free. Skills support activities that create a desired, measurable effect. A skill must contain preconditions for execution and expected effects of the application of the skill. An example of a skill would be grasping or moving a part. Skills may be composed in order to create a task$_{ii}$ like picking a part from a tray, which employs the skills of part grasping and movement. Finally, an application composes multiple tasks$_{ii}$ in order to perform a high-level activity, such as assembling several parts to form a finished product.

A planning system composes the robot tasks$_{ii}$ to perform complex assembly activities that may be altered on a run-by-run basis. During execution, skill preconditions and effects can be used to perform error checking and assembly verification. Further development of this use case will include examining the exact content that is required to specify behaviors, skills, tasks$_{ii}$, and applications along with their preconditions and effects. By standardizing these data elements, planning systems will be able to use tasking libraries provided by different vendors to program a robot to perform agile assembly operations.

Purchasing Laboratory Equipment

Pharmaceutical companies and research laboratories frequently purchase equipment. This requires exchange of information with suppliers about the equipment, its capabilities, and its intended use. Unfortunately, the representation of such information is not sufficiently standardized which may lead to miscommunication.

For example, the Robot Scientist "Eve" from the University of Manchester is designed to carry out automated drug discovery investigations. It is a fully autonomous robot that performs experiments with yeast (*Saccharomyces cerevisiae*) assays that are fast and amenable to automation [26]. Recently, Eve joined a high-profile Big Mechanism program that focuses on cancer research (*www.darpa.mil/program/big-mechanism*). In order to integrate Eve with Big Mechanism, the team must extend and align Eve's capabilities to include experimenting with human cells. This requires purchasing new equipment and integrating the new devices into the Eve robotic system.

It is beneficial to provide our potential suppliers with the specification of tasks$_i$ Eve needs to work on and the required robotic behavior. Eve has to deal with a variety of tasks$_i$ at different levels of granularity: testing research hypotheses, planning and scheduling experiments, managing consumables, etc. up to trashing used pipettes. In the absence of an agreed standard for the representation of these tasks$_i$, the specification of Eve is supported by an in-house Eve equipment ontology and the EXACT (experimental actions) ontology [21]. EXACT defines such typical experimental steps as *move, add, incubate*, and *measure*, along with their key properties such as *entity* (i.e., what is being incubated), *period* (i.e., of shaking), and *temperature*.

In order to accomplish her goal, Eve needs to carry out a "cherry picking" procedure that requires the capability of selective picking pre-defined chemicals from a large library and adding them to biological samples at different, preferably very small volumes, to enable high throughput. Such an experimental procedure (i.e., *robotic behavior*) corresponds to a sequence of tasks$_{ii}$ (defined in EXACT as actions). An example of a task$_{ii}$ is:

- add entity 1 to entity 2
- entity 1: honokiol, daidzein, fulvestrant, etc.
- volume: 20 nL–100 nL
- entity 2: cell culture
- container: wells in 384 well plate.

We represented desired behaviors using actions$_{ii}$ defined in EXACT; asked potential suppliers to provide us with the specification of their equipment that has the functionality to fulfil our requirements; and selected the most suitable ones. For the considered example, the Echo 550 liquid handler was selected as it can transfer very small volumes using acoustics, and this capability is essential for high throughput experimentation.

However, the task$_{ii}$ as defined in EXACT did not support specification of the necessary integration with EVE, so the ven-

dors misunderstood the requirements and the system selected did not provide the necessary access to their proprietary software requiring additional negotiations to support integration with EVE. If one assumes that CORA and the Task Ontology are widely accepted standards, then we will be able to communicate in a standardized way and our purchase requests should be processed accurately and meaningfully. Adoption of the Task Ontology as the standard for the description of desired uses of robotic equipment will significantly improve the exchange of information between different parties, for example between suppliers and buyers of laboratory robotic equipment.

Automotive Cable Industry

Robotic manufacturing systems are widely used in the automotive industry. However, in specific segments such as the automotive cable manufacturing industry, robots have not been successfully deployed. The main reason for this issue is the complexity and number of tasks$_i$ that must be performed. Robot cells do exist to seal automotive plugs on car cables, but their programming is not adequate for an agile manufacturing system. The work cell program must be developed from scratch for each new plug, meaning that hundreds of sealing patterns can exist in the factory database.

The task ontology driven framework defines the tasks needed to enable the robotic cell to generate a given seal pattern. An ontology was proposed [7] based on the IEEE 1872-2015 standard [1, 19] and built from the robot, environment, process knowledge, and knowledge model. The standard was used to define the machine components representation, and the task ontology will address the processes and environment at the application domain level.

The real application scenario consists of the placement of two different types of seals in a cable plug terminal with 56 holes [7]. This isolates those which do not receive electric cables. As such, the robot has two different types of capabilities, one for each type of seal insertion.

Each of the different sealing patterns requires several robot tasks$_{ii}$ to be completed in order to achieve the production plan goals$_i$. Within the Task Ontology terminology above, the goal$_i$ can be defined as 'Insert seals in a predefined pattern into the cable plug terminal.'

At the Process branch, as depicted in Fig. 5 the needed RobotBehavior required is motion. As such, the RobotMotion class defined in CORA is instantiated to accomplish the dependent tasks.

The Tasks defined in Fig. 5 are directly related to the selection of actuators. For example, the tasks *ReleaseSeal* and *CaptureSeal* use the pneumatic actuator, where two Subtasks were defined for each, e.g., '*MoveInnerPneumatic*'. The task *RotateBase120* use a 3-phase electric motor to rotate the base where the cable plug terminals are to be located for seal insertion. All the other tasks use the two Direct Current (DC) motors of the robot joints to move the robot in the XY plane.

FIGURE 5. Snapshot of the task ontology driven framework [7], related to the High Level Robot Motion Processes Branch.

SUMMARY AND NEXT STEPS

Factories of the future should comprise robot systems able to efficiently communicate and coordinate with each other and with humans. Towards the fourth industrial revolution, ontologies should provide a capable context to describe and drive human–robot and robot–robot interactions. In general, ontologies can make all pertinent knowledge about a robots' tasks available, thus promoting mass computerization of production lines.

While providing comprehensive axiomatization of the whole domain of Robotics & Automation is impractical, a partial axiomatization is beneficial. IEEE CORA was the first step in that direction, providing the initial scaffolding upon which new ontologies can be built. The next step is the creation of a task-level domain ontology, as initiated by the IEEE Robot Task Representation group. The foundations of this new ontology, the structure, and the development approach are described in this paper, as the basis for building the ontology. The need for this ontology has been demonstrated both through the differences between the current terminology in the use cases and proposed terminology and through the provided use cases.

DISCLAIMER

Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the software tools identified are necessarily the best available for the purpose.

REFERENCES

[1] IEEE Standard Ontologies for Robotics and Automation. *IEEE Std. 1872-2015*, 2015.

[2] S. Balakirsky. Ontology based action planning and verification for agile manufacturing. *Robotics and Computer-Integrated Manufacturing*, 33:21 – 28, 2015. Special Issue on Knowledge Driven Robotics and Manufacturing.

[3] S. Balakirsky and Z. Kootbally. An ontology based approach to action verification for agile manufacturing. In *Robot Intelligence Technology and Applications 2*, pages 201–217. Springer, 2014.

[4] G. Bruno. Measuring product semantic similarity by exploiting a manufacturing process ontology. In *Industrial Engineering and Systems Management (IESM), 2015 International Conference on*, pages 1251–1257. IEEE, 2015.

[5] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins. The ontology of tasks and methods, 1998.

[6] A. Chibani, A. Ferrara, T. Facchinetti, S. R. Fiorini, P. Goncalves, J. I. Olszewska, A. Olivares-Alarcos, V. Ragavan, S. Redfield, B. Spencer, E. Prestes, and H. Li. Ontology for autonomous robots. 2016. Under review.

[7] R. Farinha and P. Gonçalves. Knowledge based robotic system, towards ontology driven pick and place tasks. *Romanian Review Precision Mechanics, Optics and Mechatronics*, 49:152–157, 2016.

[8] M. Fernández, A. Gómez-Pérez, and N. Juristo. METHONTOLOGY: From ontological art towards ontological engineering. Technical Report SS-97-06, AAAI, 1997.

[9] L. A. Fraser. *A Hierarchical Control Architecture for Robust and Adaptive Robot Control*. PhD thesis, UNIVERSITY OF NEVADA, RENO, 2016.

[10] A. Freitas Martins and R. de Almeida Falbo. Models for representing task ontologies. Springer, 2008.

[11] P. J. Gonçalves and P. M. Torres. A survey on biomedical knowledge representation for robotic orthopaedic surgery. In *Robot Intelligence Technology and Applications 2*, pages 259–268. Springer, 2014.

[12] H. Goto, J. Miura, and J. Sugiyama. Human-robot collaborative assembly by on-line human action recognition based on an FSM task model. In *Human-Robot Interaction 2013 Workshop on Collaborative Manipulation*, 2013.

[13] M. Helmert. *Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition*. Springer-Verlag Berlin Heidelberg, 2008.

[14] S. Joshi and S. Jeffrey. *Computer control of flexible manufacturing systems - research and development Edition*. Springer Netherlands, 1994.

[15] G. Lortal, S. Dhouib, and S. Gérard. Integrating ontological domain knowledge into a robotic DSL. In *Proceedings of the 2010 International Conference on Models in Software Engineering*, MODELS'10, pages 401–414, Berlin, Heidelberg, 2011. Springer-Verlag.

[16] R. Mizoguchi, J. Vanwelkenhuysen, and M. Ikeda. Task ontology for reuse of problem solving knowledge. *Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing*, pages 46–59, 1995.

[17] L. Paull, G. Severac, G. V. Raffo, J. M. Angel, H. Boley, P. J. Durst, W. Gray, M. Habib, B. Nguyen, S. V. Ragavan, S. S. G., R. Sanz, M. Seto, A. Stefanovski, M. Trentini, and H. Li. Towards an ontology for autonomous robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1359–1364, Oct 2012.

[18] J. R. Peters. *Machine learning of motor skills for robotics*. PhD thesis, University of Southern California, 2007.

[19] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. 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*, 61(11):1193–1204, 2013.

[20] F. Proctor, S. Balakirsky, Z. Kootbally, T. Kramer, C. Schlenoff, and W. Shackleford. The canonical robot command language (crcl). *Industrial Robot*, 43(5), 2016.

[21] L. Soldatova, D. Nadis, R. King, P. Basu, E. Haddi, V. Bauml, N. Saunders, W. Marwan, and B. Rudkin. Exact2: the semantics of biomedical protocols. *BMC Bioinformatics*, 15(Suppl 14), 2014.

[22] M. Tenorth and M. Beetz. KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots. *Int. Journal of Robotics Research*, 32(5):566 – 590, April 2013.

[23] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz. The roboearth language: Representing and exchanging knowledge about actions, objects, and environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1284–1289. IEEE, 2012.

[24] E. G. Tsardoulias, C. Zieliński, W. Kasprzak, S. Reppou, A. L. Symeonidis, P. A. Mitkas, and G. Karagiannis. Merging robotics and aal ontologies: The rapp methodology. In *Progress in Automation, Robotics and Measuring Techniques*, pages 285–297. Springer, 2015.

[25] M. K. Uddin, A. Dvoryanchikova, A. Lobov, and J. M. Lastra. An ontology-based semantic foundation for flexible manufacturing systems. In *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*, pages 340–345. IEEE, 2011.

[26] K. Williams, E. Bilsland, A. Sparkes, W. Aubrey, M. Young, L. Soldatova, K. De Grave, J. Ramon, M. De Clare, W. Sirawaraporn, S. Oliver, and R. King. Cheaper faster drug development validated by the repositioning of drugs against negledcted tropical diseases. *Journal of the Royal Society Interface*, 2015.

[27] X. Zhu, S. J. Hu, Y. Koren, and S. P. Marin. Modeling of manufacturing complexity in mixed-model assembly lines. *Journal of Manufacturing Science and Engineering*, 130(5):051013, 2008.