Requirements for Building an Ontology for Autonomous Robots

Behzad Bayat, Julita Bermejo-Alonso, Joel Luis Carbonera,
Tullio Facchinetti, Sandro R. Fiorini, Paulo Gonçalves,
Vitor A. M. Jorge, Maki Habib, Alaa Khamis,
Kamilo Melo, Bao Nguyen,
Joanna Isabelle Olszewska, Liam Paull, Edson Prestes,
S. Veera Ragavan, Sajad Saeedi, Ricardo Sanz,
Mae Seto, Bruce Spencer,
Michael Trentini, Amirkhosro Vosughi, and Howard Li .

***Abstract*— PurposePurpose - The IEEE Ontologies for Robotics and Automation Working Group was divided into subgroups that were in charge of studying Industrial Robotics, Service Robotics, and Autonomous Robotics. This paper presents the work in-progress developed by the Autonomous Robotics (AuR) subgroup. This group aims to extend the Core Ontology for Robotics and Automation (CORA) to represent more specific concepts and axioms that are commonly used in autonomous robots.**

Design/methodology/approach - For autonomous robots, various concepts for aerial robots, underwater robots, and ground robots are described. Components of an autonomous system are defined, such as robotic platforms, actuators, sensors, control, state estimation, path planning, perception, and decision making.

Findings - AuR has identified the core concepts and domains needed to create an ontology for autonomous robots.

Practical implications - AuR targets to create a standard ontology to represent the knowledge and reasoning needed to create autonomous systems comprised of robots that can operate in the air, ground, and underwater environments. The concepts in the developed ontology will endow a robot with autonomy, i.e., endow robots with the ability to perform desired tasks in unstructured environments without continuous explicit human guidance.

Originality/value - Creating a standard for knowledge representation and reasoning in autonomous robotics will have a significant impact on all R&A domains, such as on the knowledge transmission among agents, including autonomous robots and humans. This tends to facilitate the communication among them and also provide reasoning capabilities involving the knowledge of all elements using the ontology. This will result in improved autonomy of autonomous systems. The autonomy will have considerable impact on how robots interact with humans. As a result, the use of robots will further benefit our society. Tedious tasks that currently can only be performed by humans, will be performed by robots, which will further improve the quality of life. To the best of our knowledge, AuR is the first group that adopts a systematic approach to develop ontologies consisting of specific concepts and axioms that are commonly used in autonomous robots.

I. Introduction

In the beginning of 2015, the IEEE-RAS Ontologies for Robotics and Automation Working Group (IEEE ORA WG) published the IEEE 1872-2015 standard, the first-ever standard elaborated by the IEEE Robotics and Automation Society. This standard defines a set of ontologies related to Robotics and Automation (R&A), chief among those being the Core Ontology for Robotics and Automation (CORA), which specifies the main and most general concepts and axioms that permeate the R&A domain. Due to the importance of this achievement, in December of 2015, IEEE ORA WG was the recipient of the *Emerging Technology Award*, a prize given annually by the IEEE Standards Association[1].

IEEE ORA WG started as a study group in early 2011, and became an official working group in November of 2011. It was comprised of several members from a cross-section of industry, academia and government that represent over twenty countries. Since the beginning, IEEE ORA WG was divided into different subgroups which were each in charge of studying a specific R&A subdomain, like Industrial Robotics, Service Robotics and Autonomous Robotics (Paull et al. 2012). Even having all these subgroups, all efforts were concentrated in the CORA development. Currently, after reaching this big step, these subgroups are focusing their activities in their respective subdomains.

Nowadays, Autonomous Robotics (AuR) is elaborating a petition that will be submitted to the IEEE RAS Standing Committee for Standards Activities (RAS-SCSA) to officialize its activities. This initial petition is a solicitation to officialize AuR as a study group and, according to the group progress, another petition will be submitted to the same committee to officialize AuR as an official working group, following the same steps of its parent group, IEEE ORA WG. These steps are necessary and required to conduct the development of any standard sponsored by IEEE RAS. This paper presents the work in-progress developed by one of these subgroups called AuR. This group aims to extend the CORA to represent more specific concepts and axioms that are commonly used in the Autonomous Robotics. Therefore, AuR is performing a wide study in different R&A domains (e.g. flying robots, mobile robots, field robots, marine systems) to identify the basic components in terms of hardware and software that are necessary to endow a robot (or a group of) with autonomy, i.e., endow robots with the ability to perform desired tasks in unstructured environments without continuous explicit human guidance. As a long-term goal, AuR targets to create a standard ontology that specifies the

[1]For more information see http://standards.ieee.org/develop/awards/etech/

domain knowledge needed to create autonomous systems comprised of robots that can operate in the air, ground, and underwater environments.

Creating a standard ontology in Autonomous Robotics will have a huge impact directly or indirectly on all R&A domains. The main benefit of a domain ontology is to set standard definitions of shared concepts identified in the requirement phase as well as to define appropriate relations between the concepts and their properties. Well-founded ontologies embed the domain terminology in both semantic and logical frameworks, which allows one to build a formal theory of the domain. This theory provides a much harder limit to the possible interpretations of the terms in the domain, constituting a preferable tool for standardization work than simple lists of term definitions written in natural language. Furthermore, an ontology may serve further purposes. In the tradition of symbolic Artificial Intelligence, a domain ontology provides a clear set of symbols to be used in reasoning mechanisms for autonomous systems, such as classification, inference and planning. The domain structure encoded in ontologies may provide the blueprint for APIs in domain-specific software packages, or the model for databases. Another relevant use of ontologies is in agent communications, where it provides vocabulary and clear semantics. For instance, future unmanned systems will need to work in teams and communicate with other unmanned vehicles to share information and coordinate activities. There is an increasing demand from government agencies and the private sector alike to use Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), and Autonomous Underwater Vehicles (AUVs) for tasks including homeland security, reconnaissance, search and rescue, surveillance, data collection, and urban planning. A standard ontology in Autonomous Robotics is the right tool that provides the underlying semantics of the vocabulary employed in problem solving and communications for such heterogeneous autonomous systems.

Previous approaches used to represent knowledge for R&A using ontologies include works related to navigation (Bateman & Farrar 2005), workspaces (Krieg-Brückner et al. 2005), knowledge representation and action generation (MacMahon et al. 2006, Coradeschi & Saffiotti 2001), route instruction (Lauria et al. 2002) and data representation (Soldatova et al. 2006). Regarding specifically Multi-Agent Systems (MAS), ontologies are already being used in such projects as Robot Earth European project (Waibel et al. 2011), Proteus project (Lortal et al. 2011), SWAMO NASA project (Witt et al. 2008), A3ME (Herzog et al. 2008), and so on.

These studies are very interesting and have represented a starting point for our work, but these ontologies are at a lower level of knowledge representation. They focus more on the description of the capacities of mobile agents than on the high level service representation for autonomous agents as we aim to do. In our work we are focusing on specifying the knowledge from each of the subjects or specific fields highlighted in Figure 1. In addition, the communication

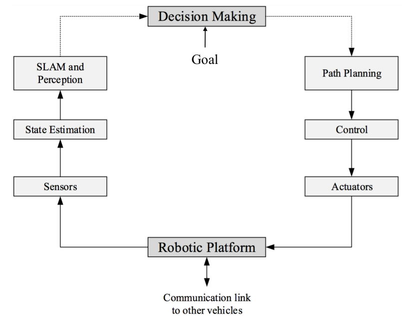

Fig. 1: Example of the structure of an autonomous vehicle system.

between autonomous agents should be explicitly defined to promote the cooperation, coordination, and communication of multiple UAVs, UGVs, and AUVs. All these elements will be discussed in this paper.

The need for ontology will be further motivated in Section II. Background information about IEEE 1872-2015 will be given in Section III. Various concepts of autonomous robots will be presented in Section IV, including robotic platforms, physical systems, modeling, autonomy, and MAS. Finally, conclusions will be given in Section V.

II. The Need for Ontologies

A. Why Ontologies? Desirable Features

Knowledge models constitute the basic component of knowledge-based approaches in fields such as artificial intelligence (AI) and robotics. According to (Smith & Welty 2001), until recently, there was a tendency to develop knowledge models that represent the knowledge in a way most suitable for performing a given task. As a consequence, the resulting knowledge models were characterized by high levels of arbitrariness, low potential reuse in other tasks, and low agreement with other knowledge models, thus undermining semantic interoperability. However, the high cost of developing knowledge models motivated the development of reusable knowledge models. Moreover, the necessity of cooperation among different stakeholders motivated the development of knowledge models that represent a common view of the reality. Due to this, in the last years there has been an increasing adoption of ontologies, since they are built for meeting these requirements.

In general, an ontology is considered as a formal and explicit specification of a shared conceptualization (Studer et al. 1998). According to this view, the conceptualization specified by an ontology includes the concepts related to the types of entities that are supposed to exist in a given domain, according to a community of people. The concepts in an ontology are, in general, the concepts that are shared by most of the community. Thus, we can say that an ontology captures a common understanding, or the consensual knowledge, about the domain. Due to this, ontologies can be used for promoting the semantic interoperability among stakeholders, because sharing a common ontology is equivalent to sharing a common view of the world. In addition, it is important to notice that, in an ontology, the specification of the conceptualization should be formal and explicit. This means that it is necessary to specify explicitly the meaning of the concepts that are included in an ontology, and this specification should be performed in a formal way, *i.e.* in a machine processable way. This ensures that the meaning of every concept should be rigorously specified and can be analyzed by humans and machines. Moreover, it is important to notice that, in general, ontologies capture the knowledge about the domain in a way that is independent of the task that would use this knowledge. Due to this, ontologies can also be viewed as reusable components of knowledge.

B. Why Ontologies? Lessons from Software Engineering

R&A systems are increasing in complexity. However, even today there are no generally accepted methods or general design frameworks or tools for the design of complex robotic systems (Ragavan & Ganapathy 2007). A tried and tested method in the software development community for handling design complexity is to increase the level of abstraction and modularity. Object-Oriented methods, Middleware, and Component Based Development have partially addressed design and development issues such as abstraction, modularity, integration, and reuse. Where Object-Orientation failed (Gabriel 2002, Astrachan et al. 2005), other approaches like Service Orientation and Model Driven Development (MDD) succeeded and were seen as the next paradigm shift to address rest of the design issues like separation of concerns, abstraction, and modularity. The next obvious step was to shift the focus of Design and Development activity from Objects and Programming to modeling.

The paradigm shift from “everything is an object” to “everything is a model” helped solve some of these abstraction bottlenecks. Object Management Group (OMG) embarked on an ambitious Model Driven Architecture (MDA) project (based on MDD) that has helped automate design tasks and established OMG as a main driver of standardization in the Model Driven Development arena. The MDA approach is a meta-modeling paradigm that is based on four levels of model abstraction (M0 to M4) and three levels of platform independence: Computationally Independent Model (CIM); Platform Independent Model (PIM); and Platform Specific Model (PSM). OMG’s grand vision states that by automatic transformation in steps, an executable implementation results and will lead to complete design automation. However, in reality, there are several issues (Ragavan et al. 2015). MDA has helped automate transformation from PIM to PSM to Code, but CIM to PIM transformation is done manually. The task of defining transformations using the UML metamodel is neither easy nor error-free (France et al. 2006). CIM frameworks and contents of CIM corresponding to the notion of computational independence are rarely discussed (Kherraf et al. 2008) and practical implementations have many pitfalls (Kleppe et al. 2003). Model transformations are still rare, and they focus more on the data levels than they do on the knowledge levels.

A major bottleneck is the lack of semantics and formal knowledge representation in MDA frameworks (Kleppe et al. 2003). Ontologies can help overcome these bottlenecks, providing a structure and framework for design along with a formal representation of knowledge models, facilitating technology reuse. As the design is based on ontologies, its level of abtraction is raised, eliminating language specific constraints in which design artifacts are expressed, while allowing reasoning capabilities to be integratd within the designed system. Hence, we argue that the next logical step is to lift the level of Design and Development tasks from objects and component models to ontology-based knowledge models. In the future, we hope, this will lead to ontology-based design and development and contribute to the emerging domain of Ontology-Based Software Engineering.

III. IEEE 1872-2015 AND CORA

IEEE RAS ORA WG (Schlenoff et al. 2012*a*) worked from 2011 to 2015 to develop a set of standard ontologies in R&A. Its work resulted in the *1872-2015 - IEEE Standard Ontologies for Robotics and Automation*, approved by members of the community. IEEE 1872-2015 defines a set of ontologies aimed at formalizing some central notions in R&A. The main ontology in this set is CORA which specifies concepts and relations that are core to the whole field (Schlenoff et al. 2012*a*, Prestes et al. 2013, Carbonera et al. 2013, Fiorini et al. 2015). One of its objectives is to serve as a basis for future ontology-development efforts, such as the one described in this article.

CORA was founded in the Suggested Upper Merged Ontology (SUMO) (Niles & Pease 2001*a*), a top-level ontology providing some elemental notions such as physical object, process, and regions. Its development is drawn heavily from methodologies based on formal ontologies, such as ONTO-CLEAN (Guarino & Welty 2004). CORA has three major concepts to describe robotic entities: robot, robot group, and robotic system. In CORA, a robot is essentially an agentive device that can act on its own or under control of another agent. CORA tries to be as inclusive as possible in what can be considered a robot; it refrains from defining sufficient conditions for robot. A robot group is a “group of robots organized to achieve at least one common goal.” This concept generalizes entities such as robot football teams and complex robots formed from many individual robots. Finally, a robotic systems is a system including robots and a supporting environment.

IEEE 1872-2015 also includes three other ontologies:

- **CORAX** — defines some notions that are useful for R&A, but too general to be included in CORA, such as design, physical environment, interaction, etc.;
- **RPARTS** — defines the different roles that component parts might have in a robot;
- **POS** — defines general notions associated with spatial knowledge (position and orientation represented as points, regions and coordinate systems)

CORA and other ontologies in IEEE 1872-2015 are too broad to be used directly in any system implementation.

Roboticists and ontologists are expected to extend them in specific application domains. Recently, some efforts have been made in that direction. This paper is an example, focusing on autonomous robots. The ontologies developed by our group shall comply with IEEE 1872-2015. In this context, compliance means that axiomatic definitions in our ontologies must be consistent with the standard, particularly regarding subsumption relations.

IV. Autonomous Robots

Significant research is in progress to support autonomy for MAS consisting of AUVs, UGVs, and UAVs. We have contributed to these efforts by investigating fundamental issues in autonomous robots. In order to develop the ontology for autonomous robots, various components of autonomous robots need to be investigated. In this section, subsystems and components of an autonomous robot shown in Figure 1 are described in detail. Subsystems and components of an autonomous robot described in this section are shown in Fig. 2.

A. Robotic Platforms

AUVs consist of a platform, sensors, control fins, propellers, front-seat and backseat computers, navigation system, control system, communication system, and base station. Autonomous UAVs consist of an airframe, sensors and actuators, state estimator, stabilization control system, autopilot, navigation system, automatic heading reference system, firmware, communication link, and ground control station. An autonomous UGV consists of a platform, mission computer, actuators, sensors, control system, navigation system, datalink, and base station. This section will summarize these three platforms.

1) Unmanned Underwater Robots: The development of AUVs started in early 1970s (Paull et al. 2013, Paull et al. 2014, Saeedi, Seto & Li 2015). Advancement in the computational efficiency, compact size, and memory capacity of computers in the past 20 years has accelerated the development of AUVs (Li et al. 2010, Paull et al. 2010). As decision making technologies evolve towards providing higher levels of autonomy for AUVs, embedded service-oriented agents require access to higher levels of data representation. These higher levels of information will be required to provide knowledge representation for contextual awareness, temporal awareness and behavioral awareness. In order to achieve autonomous decision making, the service oriented agents in the platform must be supplied with the same level of knowledge as the operator. This can be achieved by using a semantic world model and ontologies for each of the agent's domains. More details about the work developed by our Working Group are reported by Miguelanez in (Migueláñez & Patrón 2012).

2) Unmanned Aerial Robots: UAVs can be viewed as general platforms where systems and components can be aggregated to provide functionalities required to be performed in a complex mission (Nagaty et al. 2013, Nagaty, Thibault, Trentini & Li 2015, Saeedi, Thibault, Trentini & Li 2015, Nagaty, Thibault, Trentini, Facchinetti & Li 2015). A mission has several tasks that are required to be performed according to a sequence, based on a plan which evolves rapidly and dynamically in the case of collaborating agents. The agents are distributed and real-time communication is a major requirement. For example, an unmanned aerial vehicle must be capable of establishing communication with a ground station to execute some tasks such as map building, motion planning and telemetry monitoring among others. Some of these tasks must be performed on-board the UAV. To perform motion, a key capability of a UAV is to determine its pose in an unknown environment, which is estimated by fusing the data from several different sensors, such as: gyroscope, accelerometer, barometer, GPS, temperature sensor, visual sensor. In addition to other common application, ontologies can be used for task and mission planning. Major challenges are the knowledge representation of planning problems that are generally encountered by the UAV community.

3) Unmanned Ground Robots: To perform tasks efficiently, UGVs must process not only low-level sensor-motor data but also high level semantic information. The data and information are bidirectionally linked, with the low-level data passed upwards and the high-level information returned downwards. Knowledge needs to be represented and defined in order to be integrated.

For UGVs, the sub-systems that have been identified for knowledge representation are detailed in (Siegwart et al. 2011).

B. Subsystems and Components

An autonomous robot is equipped with sensors, actuators, communication systems, and embedded computing systems. The *standard IEEE 1872-2015* (Section III) describes general notions of these components. These concepts must be specialized for autonomous robots, as well as, complemented by new concepts.

1) Sensors and Actuators: A sensor measures a physical property (mechanical, thermal, magnetic, electric, optical, etc.) that will be altered/changed by an external input stimulus to produce, in general, an electric signal. Selection criteria of sensors depend on many factors, such as their ability to fulfil the requirements, their signal characteristics, whether they are active or passive, their availability, power consumption, environmental conditions, shape and size, cost, etc. Many types of sensors are used in robotics to increase agility, enhance intelligence and decision making capabilities. Sensing creates the awareness of something via sensors while perception is the process of acquiring, interpreting, selecting, and organizing sensory information. This may require integrating information from different sensors (also known as sensor fusion).

An actuator's primary function is to convert energy into work. If the actuator only receives electric, pneumatic or hydraulic energy and converts energy to power/work, and results in its behavior (e.g. motion) it is called a transducer. If the work or power output of the transducer needs to be varied, a power control component is needed, which in turn needs a physical measure which is received as feedback from a sensor component (e.g. encoder). When all these components are aggregated as a module, to fulfil a function, it can be said to provide a service. The aim of each service is to realise one or more system function(s). [2] Each operational function may require one or more parameters to fulfill its requirements and establish its control capabilities. Many types of actuators have been developed using different design, technological and operational principles. In robotics, actuators are associated with actions that affect processes and working environments. The robot can use such control actions to achieve perceptual and specific goals. In wide range of application requirements, actuator actions are controlled based on control algorithms and sensor information to enhance its functional and control capabilities.

Actuator and sensor definitions are not as straightforward as mentioned above. Different perspective emerge if viewed as a component/module or system. If servo motor is viewed as a separate component, then encoder's output is available to the servo controller, If motor-encoder-motor driver is viewed as a module, it is not available beyond the controller level. A DC drive in this case servo motor is a module consisting of a sensor and transducer component.

2) Embedded Controllers: Thus, computing hardware and software components play a central role in the implementation of logics and behaviors of autonomous robots. Embedded systems are the reference domain for computing architectures in modern robotics applications. In terms of size and complexity, hardware components can range from small Micro-Controller Units (MCUs) to full-featured computers. MCUs have limited computing capacity, but are equipped with several interfaces to connect sensors, actuators, and external peripherals in general. They are thus used for the low-level interfacing with such devices. Embedded computers are more powerful. They typically run OSes, such as Linux, but they have limited support for direct interfacing with sensors and actuators. Common architectures in complex applications include several MCUs and computers. In this view, computing devices form a distributed networked system of coordinated units (Kopetz 2011). Software components include the OS and the application middleware. The OS manages the interaction with the hardware through device drivers. Moreover, it provides support for the real-time execution of application tasks (Buttazzo 2011). Real-time scheduling allows to predictably run multiple concurrent tasks on the computing platform, achieving the timing constraints of each task. The application middleware is a framework that facilitates the implementation of a complex distributed application, involving many interacting components and logics. The Robot Operating System (ROS) (Quigley et al. 2009) is an example of middleware.

3) Communication Links: Robots need to communicate with each other for fulfilling multi-agent missions. Different communication topologies are suggested for robot communication such as star, tree, ring, mesh, line, bus, and hybrid topologies. Further, there are also different possible models of communication, e.g. they can have master/slave, rotating token, etc. Determining the appropriate communication system depends on mission circumstances and communication hardware and limitations, such as distance of units from each other, communication media and technologies, bandwidth limitations, delay, noise, communication speed, etc.

C. System Modeling

System models are used for state estimation of autonomous robots. A system model is a mathematical relation which defines the behavior of a given system. A robot model describes the changes to a robot's state that are expected to occur due to the actuator commands and external disturbances. The models cannot describe all properties of real systems since it is not practical to model all physical effects.

1) Coordinate Systems: Coordinate systems are used for stating the position and orientation of a robot system. There are several standard coordinate systems that can be used for describing system position and orientation. In general, 6 Degrees Of Freedom (DOF) are required to describe the position and orientation of a robot in 3D.

2) Kinematics Models: The kinematics model describes the mechanics concerned with the motion of an object without reference to the forces or torques that cause the motion. A kinematic model is stated as a mathematical formula which can be either linear or non-linear depending on the robot structure and assumed coordinate system.

3) Dynamic Models: A dynamic model represents the motion and behaviour of an object over time with reference to the force and toque that cause the motion. A dynamic model accounts for changes in the state of a system over time, while a static model describes the system in equilibrium. Normally, dynamic models are represented by differential equations.

D. Autonomy

For the proposed ontology, the AuR sub-group has been working on path planning, perception, and control modules for air, ground and underwater vehicles to devise formal knowledge representations as well as reasoning capabilities for autonomous robots.

1) Architecture and Decision Making: Autonomous systems can be based on different architectures with different levels of complexity ranging from a simple reactive architecture, to a deliberative architecture to a cognitive architecture. Reactive systems are based on simple Sense-Act loops while deliberative systems employ more sophisticated Sense-Decide-Act loops as illustrated in Figure 3 to endow the system with reasoning and decision making capabilities.

More sophisticated autonomous systems can be built based on a cognitive cycle of Sense-Aware-Decide-Act-Adapt-Learn that extends the deliberative cycle of Sense-Decide-Act by adding situation awareness, adaptation and learning capabilities as illustrated in Figure 4. In these systems, the

[2] A module in this context means a physical entity that has a one-to-one correspondence with a function. Service is a self-contained functional unit which when aggregated provides the overall functionality of the device.

collected data is processed to achieve different levels of situation awareness such as the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future and possible consequences. Decision making modules are responsible for making a decision in the absence of certainty. This uncertainty, if not handled correctly, may result in wrong beliefs about its state and/or the state of the environment. An adaptation module endows the system with the ability to adapt its behavior based on the context extracted from the situation awareness modules.

This adaptation can take the form of changing role assignments or re-planning to cope with the changing environment and unexpected events such as agent failure or communication degradation. Adaptive systems can also control the use of the available sensing and acting resources in a manner that synergistically improves the process of data gathering and ultimately enhances situation awareness and decision making. For example, a system can selectively collect data based on a specific context of interest. This context can be any information that can be used to characterize the situation. This context-aware data gathering process allows acquiring only the truly necessary and relevant data that reduce uncertainty or wrong beliefs about system state and/or environment state. These wrong beliefs usually result in wrong actions.

Learning from previous experience is a crucial feature of a cognitive system. A learning module can be designed to endow the system with the ability to learn new tasks or to automate repetitive ones and to provide smart guidance to a human whenever needed.

2) Sensing, Estimation, and Target Engagement : Nowadays robots are moving from structured to unstructured environments, to work closely with humans in daily real world tasks. Examples can be seen in aerial, marine and land robots, with applications in inspection, surveillance, and so on. Computer Vision plays a decisive role in the autonomy of state-of-the-art robots, because of the low cost and availability of sensors, and the algorithms available to extract information from images. Numerous methods are available from the computer vision, image processing, and related fields of research.

Various works on ontologies were developed (Compton et al. 2012, Maillot et al. 2004), mainly focused on computer vision and image processing applications or on automated scene understanding (Olszewska 2012). Review papers were presented (Fiorini & Abel 2010, Schlenoff et al. 2013). Although some works tried to join computer vision and robotic ontologies (Gonçalves et al. 2015) in specific scenarios, none of them completely tackled the broader interaction with robots, and none of them obtained a full ontology that takes into account computer vision and image processing for robotics tasks. It is our goal to develop such an ontology within the autonomous robotics ontology. Such a tool can be effectively developed under an ontological framework. The main characteristics of the robotic vision ontology, under the IEEE Core ontology for Robotics and Automation (Schlenoff et al. 2012*b*) are to be aligned with the following:

- IEEE SUMO upper ontology (Niles & Pease 2001*b*),
- IEEE Core ontology for Robotics and Automation (Schlenoff et al. 2012*b*),
- SSN Ontology of the W3C semantic sensor (Compton et al. 2012)

One of the challenges with unmanned systems is to determine the intent of the targets as well as to clearly establish who is responsible for the actions of the unmanned systems. Depending on the scenarios, the defence can employ a different rule of engagements. For example, it is possible to design an area where no targets are allowed. In this case, as soon as targets enter this area, they are considered as enemies and the defence will engage them with appropriate weapons. It is also required to be able to override the actions of unmanned systems by a human operator. This kind of rule of engagements is described by (Klein 2003).

3) Perception and Simultaneous Localization and Mapping: Simultaneous Localization and Mapping (SLAM) is a process which aims to localize an autonomous mobile robot in a previously unexplored environment while constructing a consistent and incremental map of its environment (Saeedi et al. 2014*a*, Saeedi et al. 2014*b*, Saeedi et al. 2012*a*, Saeedi et al. 2012*b*, Saeedi et al. 2016, Saeedi, Trentini & Li 2015, Saeedi, Paull, Trentini & Li 2015).

The IEEE Robot Map Data Representation Working Group is currently working on the standard for map representation.

4) Path Planning: Path planning can be used to solve coverage and navigation problems (Choset 2005):

1) Navigation - finding a collision-free path through an obstacle-laden environment.
2) Coverage - passing a sensor over every point in the environment.
3) Localization - Using sensor data to determine the configuration of the robot within the environment.
4) Mapping - Using a sensor to explore a previously unknown environment.

5) Control: The control and navigation functionalities are essential elements for autonomous robots to be able to execute the desired missions and paths accurately. An application of special interest is the autonomous vehicle navigation (AVN). AVN controllers are typically organized in cascade, as depicted in Figure 5. The highest level (level 4) is the motion planning and the trajectory generation. With the information provided by the motion planning, guidance control algorithms based on translational (kinematic/dynamic) models are normally executed at level 3 to perform path tracking or path following. At level 2, dynamic/stabilization control loops are performed. This comprises lateral and longitudinal dynamic control in the case of wheeled mobile robots and hovercrafts, or the rotational control of aerial and underwater vehicles. At this level the goal is to keep the longitudinal and lateral velocities of the vehicle or the robot attitude and its time derivatives stabilized around an operation point against possible external forces which may disturb the system. Finally, sensor/actuator control systems are located at level 1, which are designed to directly act on the throttle, breaks, elevators, ailerons, propellers, among others.

E. Multi-Agent Systems

In order to understand the different ontological aspects of multi-robot systems (MRS), the following subsections describe multi-agent system (MAS) paradigm commonly used to model MRS, the concept of cooperation and coordination in MAS and MRS modeled as a cooperative MAS.

1) MAS: The concept of agency as a computational entity is dependent on the application domain in which it operates. As a result, there are many definitions and theories on what actually constitutes an agent and the sufficient and necessary conditions for agency. The simplest definition of an agent has been proposed by Russel and Norvig (Russell & Norvig 1995). They define an agent as an entity that can perceive and affect its environment. Wooldrige and Jennings extend this definition by considering autonomy, reactivity, proactiveness, and social ability as essential properties of this computational entity (Wooldridge & Jennings 1995). These additional features make the agent not only able to react to external stimuli, but also able to exhibit opportunistic, goal-directed behavior and take the initiative autonomously where appropriate to accomplish a given task individually or in collaboration with other agents. Similarly, Nareyek defines an agent as an entity that incorporates proactive autonomous units with goal-directed behavior and communication capabilities (Nareyek 2001). An artificial agent is a computational entity that behaves like a software robot with an ontological commitment and agenda of its own. This agent is able to perform autonomous or semi-autonomous actions alone and/or in collaboration with other agents in order to achieve an individual or common goal/s. Balch and Parker define intelligent agents are defined as computational entities which have (Balch & Parker 2002) objectives, actions, and a knowledge domain. These intelligent agents used to be designed and built using mentalistic notions such as beliefs, goals and intentions. Additionally, they are suited in an environment, and capable of flexible autonomous actions in order to fulfill their objectives. A cognitive agent is a computational model of human cognition: it perceives information from its environment, assesses situations using knowledge obtained from human experts, makes decisions and takes action to reach its goals. Although intelligent agents can simulate cooperative behavior, only cognitive agents are capable of true cooperation since they can anticipate the needs of other agents. If an autonomous robot is able to rationally cooperate with other robots, learn from experience and adapt its behavior to cope with changes in the environment, we can call it a cognitive agent.

Multi-agent systems (MAS) are systems composed of multiple intelligent or cognitive agents interacting together to achieve a common goal or solve a problem. These agents are often trying to achieve more complex objectives than they could achieve individually. Thus each agent has to have the capacity to model the actions and objectives of other agents (Balch & Parker 2002). An MAS can have one or more of the following characteristics: no explicit global control; distributed resources, expertise, intelligence, and processing capabilities; an open environment full of uncertainties and an emphasis placed on social agency and social commitments.

2) Coordination and Cooperation: Cooperation is the purposive positive interference of agents to further the achievement of a common goal or goals compatible with their own (Benaskeur et al. 2011). In MRS, the major motives for cooperation among the agents would be to increase survival capacity, i.e. maintain group functionality, to increase the performance, i.e. capacity of achieving tasks and/or to avoid or solve potential or actual conflicts (Khamis et al. 2006). Information sharing and cooperation among the agents can also enhance the shared situational awareness, which dramatically increases mission effectiveness (Khamis & ElGindy 2012). Although cooperation and collaboration are generally held to be the opposite paradigms from competition, the concept of cooperation differs from the concept of collaboration. Figure 6 depicts the relation between cooperation, coordination, collaboration and competition. Cooperative agents communicate in order to achieve better the goals of themselves or of the system in which they exist. These goals might or might not be known to the agents explicitly, depending on whether or not the cooperating agents are goal-based. Communication can enable these agents to coordinate their actions and behaviors, resulting in systems that are more coherent. Coherence is how well a system behaves as a unit. Collaborative agents are empowered politically to negotiate, as well as to advocate for, programs and policies leading to more comprehensive service delivery.

Cooperative agents in a MAS can have three main forms of cooperation based on the factors that motivate such cooperation as defined by Schmidt in (Schmidt 1990).

- **Augmentative cooperation** occurs when agents have a similar *know-how*, but they must be multiplied to perform a task that is too demanding for only one agent. Since the agents have similar know-how, the complex task is decomposed into simpler sub-tasks with similar requirements. Multiple agents perform these sub-tasks to finish the complex task. Examples of augmentative cooperation in distributed surveillance systems include, but are not limited to, target cueing/handoff (Benaskeur et al. 2011) and establishing communication through relaying.
- **Integrative cooperation** means that agents have different complementary *know-how*s and it is necessary to integrate their contribution for achieving a task. Cooperative target detection and tracking (Elmogy et al. 2010) using static and mobile sensors in a distributed surveillance system is an example of integrative cooperation.
- **Debative cooperation** occurs when agents have similar *know-how*s and are faced with a unique task, for which they seek the best solution by comparing their results.

Cooperation forms can be classified based on different criteria such as access to resources and agents' skills (Ferber 1999). In this case, we can have independence, simple collaboration, obstruction and coordinated collaboration as forms of cooperation. In independence form, there are sufficient resources and sufficient skills. In simple collaboration, there are sufficient resources but insufficient skills, which require simple addition of skills like in case of task allocation. In obstruction, there are insufficient resources and sufficient skills like scheduling scenario. Finally, in a coordinated collaboration scenario, there are insufficient resources and insufficient skills that require both task allocation and management of the limited available resources. Regardless the forms of cooperation, a coordination mechanism is required to manage the interdependency among the agents. The cooperative agents are not required to negotiate on behalf of the system they represent, while competitive entities use negotiation as a coordination strategy.

Cooperation is a fundamental part of an efficient decision making and problem solving. Effective cooperation can be facilitated by communication and coordination between cooperative agents with the aim to establish the shared understanding required to achieve shared goals. This requires having awareness of other actions, thoughts and affections with the possibility of sharing common interest. There is a need to balance between the degree of autonomy and the level of cooperation while executing tasks within environments that are either dangerous or inaccessible for humans and segmented with bottleneck communication delay (Habib 2008, Habib 2011*a*, Habib 2011*b*).

3) Multi-robot systems (MRS) as Cooperative MAS: Multi-robot systems (MRS) are a group of robots that are designed aiming to perform some collective behavior. By this collective behavior, some goals that are impossible for a single robot to achieve become feasible and attainable. MRS have been on the agenda of the robotics community for several years. It is only in the last decade, however, that the topic has really taken off, as seen from the growing number of publications appearing in the journals and conferences. One of the reasons that the topic has become more popular is the various foreseen benefits of MRS compared to single robot systems. These benefits include, but are not limited to the following (Khamis et al. 2015):

- **Resolving task complexity:** Some tasks may be quite complex or even impossible for a single robot to do. This complexity may be due to the distributed nature of the tasks and/or the diversity of the tasks in terms of different requirements. Examples of these tasks include reconnaissance, surveillance, search and rescue.
- **Increasing the performance:** Performance measures are application-dependent. As an example, task completion time can be dramatically decreased if many robots cooperate to do the tasks in parallel. Spatial and/or temporal area/object coverage can be improved using multiple robots. Moreover, in some applications, these robots can cooperate to establish an ad hoc communication relay network to improve radio coverage.
- **Increasing reliability:** Increasing the system reliability can be achieved through redundancy, because having only one robot may work as a bottleneck for the whole system, especially in critical times. When having multiple robots doing a task and one fails, others could still do the job.
- **Simplicity in design:** Having small, simple robots will be easier and cheaper to implement than having only single powerful complex robot. In complex exploration missions, several simpler robots are preferable to a monolithic single robot (Parker 1998, Singh et al. 2009).

The multi-agent system (MAS) paradigm is well-suited for use in MRS where distributed computation, concurrent processing capabilities and communication between spatially distributed robots are involved. This paradigm introduces a number of new abstractions and design/development issues when compared with more traditional approaches to software development (Zambonelli et al. 2003). The agents in MAS can interact, cooperate, coordinate and negotiate with each other if necessary to achieve their individual and/or common goals of the system.

V. Conclusion

In this paper, we have described the work of the AuR subgroup. We have described the goal of the AuR group towards the development of an ontology for autonomous robots. The needs for ontology are motivated and background information about IEEE 1872-2015 is given. Various concepts of autonomous robots are described. Although the core components for autonomous systems are described, much work needs to be done to develop the standard ontology. Axioms will be further defined in the near future. Readers are encouraged to contribute to the standardization and development of the ontology for autonomous systems.

VI. ACKNOWLEDGMENT

This work was partly supported by CAPES and CNPq, Natural Sciences and Engineering Research Council of Canada (NSERC), New Brunswick Innovation Foundation (NBIF), and by the Project UID/EMS/50022/2013, through FCT, under LAETA/IDMEC/CSI.

The authors gratefully acknowledge the financial support of their organizations. The reviewers' comments are greatly appreciated. Our thanks must also go to Francesco Amigoni, Emilio Miguelanez, Craig Schlenoff, Raj Madhavan and other members of the IEEE RAS Ontologies for Robotics and Automation Working Group: Gaetan Severac, Guilherme Raffo, Julian Angle, Aleksandar Stefanovski, Phillip J Durst, and Wendell Gray's contributions to the initial work presented at IEEE IROS are greatly appreciated.

References

Astrachan, O., Bruce, K., Koffman, E., Kölling, M. & Reges, S. (2005), 'Resolved: objects early has failed', *ACM SIGCSE Bulletin* **37**(1), 451–452.

Balch, T. & Parker, L. (2002), *Robot Teams: From Diversity to Polymorphism*, Ak Peters Series, Taylor & Francis.

Bateman, J. & Farrar, S. (2005), Modelling models of robot navigation using formal spatial ontology, *in* 'Spatial Cognition IV. Reasoning, Action, Interaction', Springer, pp. 366–389.

Benaskeur, A., Khamis, A. & Irandoust, H. (2011), 'Cooperation in distributed surveillance systems for dense regions', *International Journal of Intelligent Defence Support Systems* **4**(1), 20–49.

Buttazzo, G. C. (2011), *Hard real-time computing systems: predictable scheduling algorithms and applications*, Vol. 24 of *Real-Time Systems*, third edn, Springer Science & Business Media.

Carbonera, J., Rama Fiorini, S., Prestes, E., Jorge, V., Abel, M., Madhavan, R., Locoro, A., Gonçalves, P., Haidegger, T., Barreto, M. & Schlenoff, C. (2013), Defining positioning in a core ontology for robotics, *in* 'Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on', pp. 1867–1872.

Choset, H. M. (2005), *Principles of robot motion: theory, algorithms, and implementation*, MIT press.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Phuoc, D. L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A. & Taylor, K. (2012), 'The SSN ontology of the W3C semantic sensor network incubator group', *Web Semantics: Science, Services and Agents on the World Wide Web* **17**, 25–32.

Coradeschi, S. & Saffiotti, A. (2001), Perceptual anchoring of symbols for action, *in* '17th IJCAI Conference', pp. 407–416.

Elmogy, A. M., Karray, F. & Khamis, A. M. (2010), 'Auction-based consensus mechanism for cooperative tracking in multi-sensor surveillance systems', *Journal of Advanced Computational Intelligence and Intelligent Informatics* **14**(1), 13–20.

Ferber, J. (1999), *Multi-agent systems: an introduction to distributed artificial intelligence*, Vol. 1, Addison-Wesley Reading.

Fiorini, S. R. & Abel, M. (2010), A review on knowledge-based computer vision, Technical report, UFRGS, Brasil.

Fiorini, S. R., Carbonera, J. L., Gonçalves, P., Jorge, V. A., Rey, V. F., Haidegger, T., Abel, M., Redfield, S. A., Balakirsky, S., Ragavan, V., Li, H., Schlenoff, C. & Prestes, E. (2015), 'Extensions to the core ontology for robotics and automation', *Robotics and Computer-Integrated Manufacturing* **33**, 3 – 11. Special Issue on Knowledge Driven Robotics and Manufacturing.

France, R. B., Ghosh, S., Dinh-Trong, T. & Solberg, A. (2006), 'Model-driven development using uml 2.0: promises and pitfalls', *Computer* **39**(2), 59–66.

Gabriel, R. P. (2002), 'Objects have failed', *OOPSLA Debate November* .

Gonçalves, P., Torres, P., Santos, F., Antonio, R., Catarino, N. & Martins, J. (2015), 'A vision system for robotic ultrasound guided orthopaedic surgery', *Journal of Intelligent & Robotic Systems* **77**(2), 327–339.

Guarino, N. & Welty, C. (2004), An overview of ontoclean, *in* S. Staab & R. Studer, eds, 'Handbook on Ontologies', International Handbooks on Information Systems, Springer Berlin Heidelberg, pp. 151–171.

Habib, M. K. (2008), Distributed teleoperation and collaborative environment for robotics e-learning and cooperation, *in* 'IEEE International Conference on Industrial Informatics', pp. 1358–1363.

Habib, M. K. (2011*a*), Collaborative and distributed intelligent environment merging virtual and physical realities, *in* 'IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST)', pp. 340–344.

Habib, M. K. (2011*b*), Multi robotic system and the development of cooperative mine clearance and navigation behaviors, *in* 'International Symposium on Artificial Life and Robotics (AROB)'.

Herzog, A., Jacobi, D. & Buchmann, A. (2008), A3ME-an agent-based middleware approach for mixed mode environments, *in* 'IEEE International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM'08', pp. 191–196.

Khamis, A. & ElGindy, A. (2012), 'Minefield mapping using cooperative multirobot systems', *Journal of Robotics* **2012**.

Khamis, A., Hussein, A. & Elmogy, A. (2015), Multi-robot task allocation: A review of the state-of-the-art, *in* 'Cooperative Robots and Sensor Networks', Springer, pp. 31–51.

Khamis, A. M., Kamel, M. S., Salichs, M. et al. (2006), Cooperation: concepts and general typology, *in* 'IEEE International Conference on Systems, Man and Cybernetics (SMC)', Vol. 2, pp. 1499–1505.

Kherraf, S., Lefebvre, É. & Suryn, W. (2008), Transformation from cim to pim using patterns and archetypes, *in* 'Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on', IEEE, pp. 338–346.

Klein, J. (2003), 'The problematic nexus: Where unmanned combat air vehicles and the law of armed conflict meet', *Air & Space Power Journal, Chronicles Online Journal* .

Kleppe, A. G., Warmer, J. B. & Bast, W. (2003), *MDA explained: the model driven architecture: practice and promise*, Addison-Wesley Professional.

Kopetz, H. (2011), *Real-time systems: design principles for distributed embedded applications*, Springer Science & Business Media.

Krieg-Brückner, B., Frese, U., Lüttich, K., Mandel, C., Mossakowski, T. & Ross, R. J. (2005), Specification of an ontology for route graphs, *in* 'Spatial Cognition IV. Reasoning, Action, Interaction', Springer, pp. 390–412.

Lauria, S., Bugmann, G., Kyriacou, T., Bos, J. & Klein, E. (2002), Converting natural language route instructions into robot executable procedures, *in* 'IEEE International Workshop on Robot and Human Interactive Communication', pp. 223–228.

Li, H., Popa, A., Thibault, C., Trentini, M. & Seto, M. (2010), A software framework for multi-agent control of multiple autonomous underwater vehicles for underwater mine counter-measures, *in* '2010 International Conference on Autonomous and Intelligent Systems (AIS)'.

Lortal, G., Dhouib, S. & Gérard, S. (2011), Integrating ontological domain knowledge into a robotic DSL, *in* 'Models in Software Engineering', Springer, pp. 401–414.

MacMahon, M., Stankiewicz, B. & Kuipers, B. (2006), Walk the talk: Connecting language, knowledge, and action in route instructions, *in* '21st National Conference on Artificial Intelligence - Volume 2', AAAI'06, AAAI Press, pp. 1475–1482.

Maillot, N., Thonnat, M. & Boucher, A. (2004), 'Towards ontology-based cognitive vision', *Machine Vision and Applications* **16**(1), 33–40.

Migueláñez, E. & Patrón, P. (2012), A knowledge representation framework for autonomous underwater vehicles (AUVs), *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Nagaty, A., Saeedi, S., Thibault, C., Seto, M. & Li, H. (2013), 'Control and navigation framework for quadrotor helicopters', *Journal of Intelligent & Robotic Systems* **70**(1-3).

Nagaty, A., Thibault, C., Trentini, M., Facchinetti, T. & Li, H. (2015), Construction, modeling and control of a quadrotor for target localization, *in* '2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)'.

Nagaty, A., Thibault, C., Trentini, M. & Li, H. (2015), 'Probabilistic cooperative target localization', *IEEE Transactions on Automation Science and Engineering* **12**(3), 786–794.

Nareyek, A. (2001), *Constraint-based agents: an architecture for constraint-based modeling and local-search-based reasoning for planning and scheduling in open and dynamic worlds*, Springer-Verlag.

Niles, I. & Pease, A. (2001*a*), Origins of the ieee standard upper ontology, *in* 'Working Notes of the IJCAI-2001 Workshop on the IEEE Standard Upper Ontology', pp. 4–10.

Niles, I. & Pease, A. (2001*b*), Towards a standard upper ontology, *in* 'Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001', FOIS '01, ACM, New York, NY, USA, p. 29.

Olszewska, J. I. (2012), Multi-target parametric active contours to support ontological domain representation, *in* 'Proceedings of the 18th RFIA Conference', RFIA, pp. 779–784.

Parker, L. E. (1998), 'Alliance: An architecture for fault tolerant multi-robot cooperation', *IEEE Transactions on Robotics and Automation* **14**(2), 220–240.

Paull, L., Saeedi, S., Li, H. & Myers, V. (2010), An information gain based adaptive path planning method for an autonomous underwater vehicle using sidescan sonar, *in* '2010 IEEE Conference on Automation Science and Engineering (CASE)', pp. 835–840.

Paull, L., Saeedi, S., Seto, M. & Li, H. (2013), 'Sensor-driven online coverage planning for autonomous underwater vehicles', *IEEE/ASME Transactions on Mechatronics* **18**(6), 1827–1838.

Paull, L., Seto, M. & Li, H. (2014), Area coverage planning that accounts for pose uncertainty with an auv seabed surveying application, *in* '2014 IEEE International Conference on Robotics and Automation (ICRA)'.

Paull, L., Severac, G., Raffo, G., Angel, J., Boley, H., Durst, P., Gray, W., Habib, M., Nguyen, B., Ragavan, S. V., Saeedi, S., Sanz, R., Seto, M., Stefanovski, A., Trentini, M. & Li, H. (2012), Towards an ontology for autonomous robots, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1359–1364.

Prestes, E., Carbonera, J. L., Fiorini, S. R., Jorge, V. A. M., Abel, M., Madhavan, R., Locoro, A., Gonçalves, P., Barreto, M. E., Habib, M., Chibani, A., Gerard, S., Amirat, Y. & Schlenoff, C. (2013), 'Towards a core ontology for robotics and automation', *Robotics and Autonomous Systems* **61**(11), 1193 – 1204. Ubiquitous Robotics.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. & Ng, A. Y. (2009), Ros: an open-source robot operating system, *in* 'ICRA workshop on open source software', Vol. 3.2, p. 5.

Ragavan, S. V. & Ganapathy, V. (2007), A general telematics framework for autonomous service robots, *in* 'Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on', IEEE, pp. 609–614.

Ragavan, S. V., Shanmugavel, M., Ganapathy, V. & Shirinzadeh, B. (2015), 'Unified meta-modeling framework using bond graph grammars for conceptual modeling', *Robotics and Autonomous Systems* .

Russell, S. & Norvig, P. (1995), *Artificial Intelligence: Modern Approach*, Prentice Hall.

Saeedi, S., Paull, L., Trentini, M. & Li, H. (2015), 'Occupancy grid map merging for multiple robot simultaneous localization and mapping', *International Journal of Robotics and Automation* **30**(2), 149–157.

Saeedi, S., Paull, L., Trentini, M., Seto, M. & Li, H. (2012*a*), Efficient map merging using a probabilistic generalized voronoi diagram, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 4419–4424.

Saeedi, S., Paull, L., Trentini, M., Seto, M. & Li, H. (2012*b*), Map merging using hough peak matching, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 4683–4688.

Saeedi, S., Paull, L., Trentini, M., Seto, M. & Li, H. (2014*a*), 'Group mapping: A topological approach to map merging for multiple robots', *IEEE Robotics & Automation Magazine* **21**(2), 60–72.

Saeedi, S., Paull, L., Trentini, M., Seto, M. & Li, H. (2014*b*), 'Map merging for multiple robots using hough peak matching', *Robotics and Autonomous Systems* **62**(10), 1408–1424.

Saeedi, S., Seto, M. & Li, H. (2015), Fast monte carlo localization of auv using acoustic range measurement, *in* '2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)', pp. 326–331.

Saeedi, S., Thibault, C., Trentini, M. & Li, H. (2015), 'The cobra fixed-wing georeferenced imagery dataset', *International Journal of Intelligent Unmanned Systems* **3**(2/3), 62–71.

Saeedi, S., Trentini, M. & Li, H. (2015), A hybrid approach for multiple-robot slam with particle filtering, *in* '2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 3421–3426.

Saeedi, S., Trentini, M., Seto, M. & Li, H. (2016), 'Multiplerobot simultaneous localization and mapping: A review', *Journal of Field Robotics* **33**(1), 3–46.

Schlenoff, C., Hong, T., Liu, C., Eastman, R. & Foufou, S. (2013), A literature review of sensor ontologies for manufacturing applications, *in* 'IEEE International Symposium on Robotic and Sensors Environments (ROSE)', pp. 96–101.

Schlenoff, C., Prestes, E., Madhavan, R., Gonçalves, P., Li, H., Balakirsky, S., Kramer, T. & Miguelanez, E. (2012*a*), An ieee standard ontology for robotics and automation, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems', Springer-Verlag, Vilamoura, Algarve, Portugal, pp. 1337–1342.

Schlenoff, C., Prestes, E., Madhavan, R., Goncalves, P., Li, H., Balakirsky, S., Kramer, T. & Miguelanez, E. (2012*b*), An ieee standard ontology for robotics and automation, *in* 'Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on', pp. 1337–1342.

Schmidt, K. (1990), Analysis of cooperative work: a conceptual framework, Technical Report Rep. Risoe-M-2890, Technical report, Risoe Nat. Lab., Roskilde, Denmark, Risoe Tech.

Siegwart, R., Nourbakhsh, I. R. & Scaramuzza, D. (2011), *Introduction to autonomous mobile robots*, MIT press.

Singh, A., Krause, A., Guestrin, C. & Kaiser, W. J. (2009), 'Efficient informative sensing using multiple robots', *Journal of Artificial Intelligence Research* pp. 707–755.

Smith, B. & Welty, C. (2001), Ontology: Towards a new synthesis, *in* 'Formal Ontology in Information Systems', ACM Press, USA, pp. iii-x, pp. 3–9.

Soldatova, L. N., Clare, A., Sparkes, A. & King, R. D. (2006), 'An ontology for a robot scientist', *Bioinformatics* **22**(14), e464–e471.

Studer, R., Benjamins, V. R. & Fensel, D. (1998), 'Knowledge engineering: Principles and methods', *Data and Knowledge Engineering* **25**(1-2), 161–197.

Waibel, M., Beetz, M., Civera, J., D'Andrea, R., Elfring, J., Galvez-Lopez, D., Haussermann, K., Janssen, R., Montiel, J., Perzylo, A., Schiessle, B., Tenorth, M., Zweigle, O. & Van De Molengraft, R. (2011), 'RoboEarth', *IEEE Robotics Automation Magazine* **18**(2), 69–82.

Witt, K. J., Stanley, J., Smithbauer, D., Mandl, D., Ly, V., Underbrink, A. & Metheny, M. (2008), Enabling sensor webs by utilizing SWAMO for autonomous operations, *in* '8th NASA Earth Science Technology Conference'.

Wooldridge, M. & Jennings, N. R. (1995), 'Intelligent agents: Theory and practice', *The knowledge engineering review* **10**(02), 115–152.

Zambonelli, F., Jennings, N. R. & Wooldridge, M. (2003), 'Developing multiagent systems: The Gaia methodology', *ACM Transactions on Software Engineering and Methodology (TOSEM)* **12**(3), 317–370.

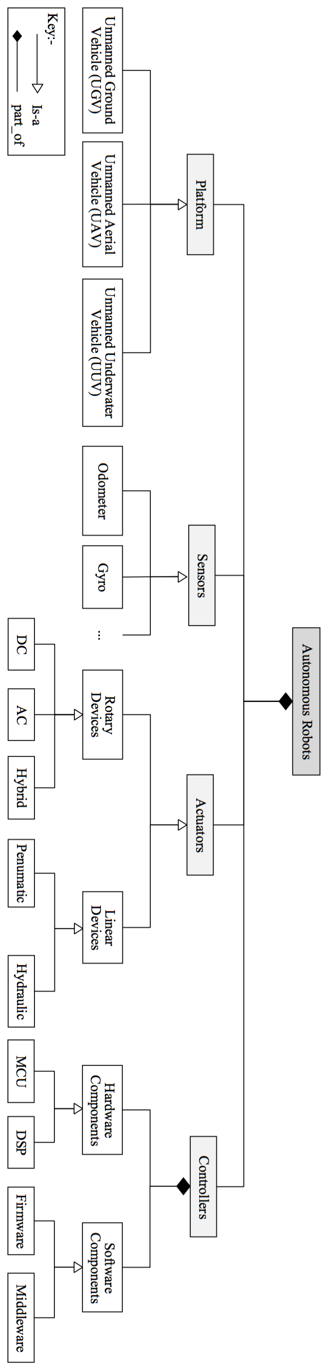

Fig. 2: Knowledge Requirements for Autonomous Robot

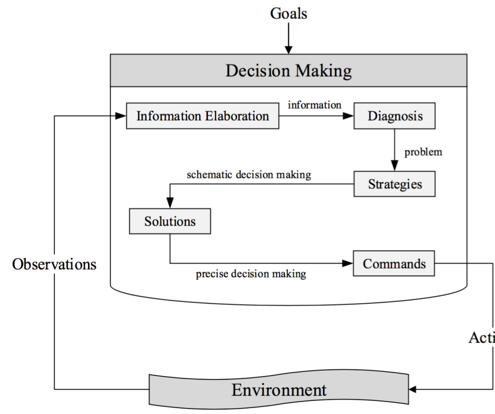

Fig. 3: Deliberative System Following Sense-Decide-Act Loop

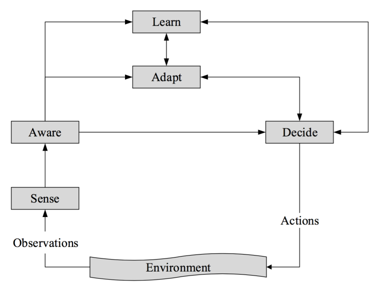

Fig. 4: Cognitive System Following Sense-Aware-Decide-Act-Adapt-Learn Loop

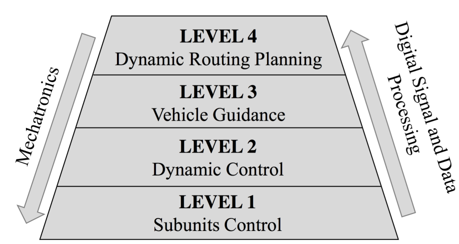

Fig. 5: Cascade-based AVN controller.

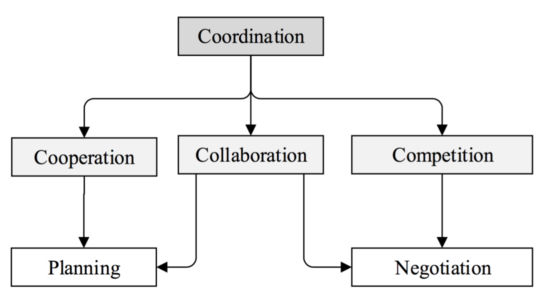

Fig. 6: Coordination and Cooperation in MRS