# APPLYING KNOWLEDGE REPRESENTATION IN FUTURE SPACE MISSIONS TO FILL MISSING INFORMATION GAPS IN APPLICATIONS

**Mehmed Yüksel[*], Marko Jankovic, and Christoph Hertzberg**

*DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Str.1 28359 Bremen, Germany,*
*name.surname@dfki.de*

## ABSTRACT

Knowledge about all phases of the life cycles of robotic systems in space is essential. Ontology as a method is a suitable way to collect and represent knowledge. This paper presents two different examples of ontologies with development and application perspectives, namely active debris removal in orbital robotics and collaborated astronaut-robot task on lunar site as future mission scenarios in space. Furthermore, we give an overview of the practical implementation to address the problem of static or dynamic knowledge in realistic application scenarios, which in our experience plays the key role in determining the added value of using ontology.

Key words: knowledge representation, ontology, active debris removal, orbital robotics, lunar robotics, task, framework, man-machine-interaction.

## 1. INTRODUCTION

This paper focuses on the application of knowledge representation in the form of an ontology at different levels and in different domains for the purposes of diverse future space missions. A mechatronical system, such as a spacecraft or a planetary robotic system, is considered a synergy of mechanical, electrical, and software components that perform various tasks. This interdisciplinary work requires a continuous, sustainable knowledge-transfer circle. Such complex systems require additional functionally specialized subsystems that perform specific tasks and thus subdivide the complexity of the system. These tasks are thus covered by the system in order to fulfill the requirements. From the perspective of a planetary or orbital robotic system, it is considered as a system consisting of mechanical components (e.g., bus/body or support structure, linkage or extremities, payloads), electrical (e.g., power sources, power management and distribution units, energy harvesting devices, computing devices, sensors, interfaces) and software components (e.g., operating system, control system, Machine Learning (ML) supported perception, navigation, and planning algorithms). In these areas, diverse knowledge is needed directly or indirectly for several purposes, e.g., classification, decision making, optimization, or adaptation. From a higher-level perspective, knowledge encompasses everything from specification and design, through development and execution, to the disposal perspective in the various life phases and technological levels of the mechatronic system. Under these different aspects, it is necessary to collect and share knowledge in order to reuse it in a reliable, consistent, and sustainable way. One way of doing it is using an ontology which can be defined as a method to collect explicit knowledge in a standardized way and generate implicit knowledge using reasoning and query functions. For this purpose, we present two application examples from our experience that address the usage of ontologies in different domains and future space missions. In addition, we give a brief overview of the state of the art of the domain. This is followed by a section describing the application of an ontology, which focuses on the development of the ontology for application-oriented design for domain experts with these two example ontologies from space domain a practical point of view. The following section describes the application of the selected ontologies for the space domain. The two example ontologies are designed to be application-oriented and are explained from the perspective of domain experts. We then lead a discussion and conclude with an outlook.

## 2. STATE OF THE ART IN THE APPLICATION OF ONTOLOGIES IN THE ROBOTICS AND ACTIVE DEBRIS REMOVAL DOMAIN

Several studies proposed ontologies to provide the required knowledge in a standardized way for a robotic domain [1]. Most ontologies remained generic and covered only partial aspects of informa-

---

[*] Corresponding author.

Email adress:mehmed.yueksel@dfki.de (Mehmed Yüksel)

tion [2, 3]. Considering the entire robot domain from design to application, ontology approaches are typically explored in the task domain and the high-level robot control and task execution perspective. From the design perspective, five main points can be identified as the following main domains. These are *robot information*, including hardware and software with robot capabilities and skills and behaviors, *robot interaction*, *task environment and workspace*, *task plan* and *temporal information*. The overlap of focus of the application domains of ontologies can be recognized in certain domains and also some domains are not yet sufficiently covered. Most ontologies from the field of robotics contain task-related environmental information. They focus more on their perspective of certain types of environments, such as indoor environments and their objects, such as household appliances and furniture, which may not be sufficient for an orbital or lunar mission. This is to be distinguished from work environments (e. g., offices) or domestic environments (kitchens, living rooms) or hospitals, factories and search and rescue areas and the associated objects and tools [2, 4]. Like in the field of human-robot interaction with intention recognition, ontology is a method that conveys domain-specific knowledge [5].

For what concerns the space debris domain, prior efforts in using an ontology to counteract the ever-growing problem of the information paradox in the field have focused almost exclusively onto the domain of space situational awareness (SSA) neglecting the active debris removal (ADR) [6]. Furthermore, most implementations appear to overlook the handling of input data from potential databases such as the ESA's Database and Information System Characterising Objects in Space (DISCOS) [7] or US Space Force's Space-Track database [8]. Cox et al. [9] present a "Space Object Ontology" designed to support the space domain awareness by enabling improved characterization of objects and related events. The authors mention integration of data from multiple data sources but do not provide additional implementation details. Similarly, Rovetto [10] illustrates an ontological architecture of the orbital debris domain. Limited details regarding the implementation method are provided within the research and the methodology used for the input of the necessary data into the ontology is not mentioned, thus hinting at a manual process. Furfaro et al. [11] describe an approach to characterize the behavior of resident space objects (RSOs) (starting from sensor measurements), classify them and execute probabilistic reasoning. The developed ontology does not include any knowledge about the ADR domain and methodology used for the input of the necessary data is not mentioned, thus hinting at a manual process. Alike, Liu et al. [12] outline an ontology for RSOs developed using expert domain knowledge and unordered machine learning rules The ADR domain was once again not within the scope of the developed ontology and the methodology concerning the input of the data into the ontology was not mentioned. Le May et al. [13], on the other hand, illustrate a knowledge graph-based method to represent RSOs and support early SSA operations and observation planning, characterized by a semi-automated data input. The graph database in question has been specifically developed with two data sources in mind: structured and unstructured. However, even this method does not model the ADR domain, thus precluding any possibility to represent it or infer knowledge from it [14].

## 3. APPLICATION OF THE ONTOLOGY IN SPACE DOMAIN

Ontology is a method of knowledge representation that aims to collect precisely semantically defined concepts with rich relationships and properties in order to obtain a common understanding and transfer it to related domains as well as between (robotic) systems [3, 15]. The application of the ontology can be summarized in four key points: (a) The ontology is used to *represent* relationships and similarities. (b) Not only to describe or search for concepts or values, but also is used to *query* semantically enriched concepts to explain not only the *"what"* but also the *"why"*. (c) Additionally, by using a reasoner on the ontology serves the implicit knowledge, which is *inferenced* depending on explicit knowledge, that includes axioms to define taxonomy, domain or range, and disjunctionality and relations. (d) Ontology servers *inconsistency check*, which finds out the contradiction cases when these axioms cause such logical inconsistencies. [16, 17]

In this paper we provide an overview of practical methods to address the problem with static and/or dynamic knowledge in two realistic application scenarios. Here we considered the knowledge from different points of view (see in Fig. 1). (a) The knowledge that is generated by ourselves is referred to as internal knowledge, which must be uniform and semantically well defined in order to be correctly interpreted and reused. (b) The external knowledge, such as the knowledge from the Internet that can be accessed via API, is called external knowledge, which is basically standardized by the application or similar. (c) The static knowledge is another point, that contains triples that define syntax such as lexicons. In contrast, (d) dynamic knowledge would then be the information that is changeable over time. The last point would then be (e) explicit or (f) implicit knowledge, which is knowledge that is known or implied after reasoning.

### 3.1. Using an ontology for active debris removal

The remediation activities or more specifically ADR is an essential tool to stabilize the existing space de-

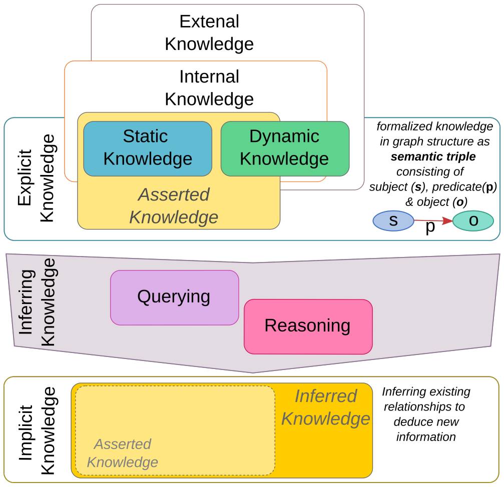

*Figure 1. A compact overview of the types of knowledge and their relationships*

bris population. Nevertheless, choosing one method over another is currently a difficult task not only due to the number of parameters characterizing targets and potential solutions but also due to the lack of coherence and structure of the available data. This has led up until now to confusion and general hesitation on how and when to best perform ADR. To bridge this gap the the onTology foR ACtive dEbris Removal (TRACER) was developed for data collection, storage and sharing of characteristics of intact derelict objects (IDOs), i. e., payloads and rocket bodies, as defined in [18], useful to ADR. The ontology defines the minimal set of physical and dynamical parameters of an object deemed sufficient to infer its most suited ADR capture method(s). Moreover, the developed software framework surrounding the domain ontology enables a transparent handling of the input data, minimizing user intervention and possibility of an error. This way, not only the management, but also the discovery of new knowledge is facilitated with an aim to make future ADR mission planning easier yet more systematic. The intended users of TRACER are ADR mission planners, domain experts and decision makers that would be able to benefit from a minimalist yet standardized way for data collection, storage and access of complex ADR domain knowledge [6, 14].

The competency questions of the developed ontology and thus, its requirements are as follows [6]:

- How could a domain knowledge about IDOs be captured in a standardized, formal, machine-interpretable way useful to ADR?
- What are the minimum parameters needed to characterize an IDO for an ADR capture phase?
- How can the degree of hazard of an IDO to an ADR capture phase be represented?
- How could the most suited ADR capture method be inferred?
- How could the input of data into the ontology be simplified and made compatible with an existing space debris catalog, such as DISCOS [7]?

The minimum classification parameters identified in TRACER as sufficient to characterize unambiguously an IDO for a capture maneuver and represented as classes are: a) attitude regime, b) onboard propellant, c) space object type, d) breakup criticality number and e) ADR capture method type. The last two parameters/classes identify the breakup risk of an object, due to the possibility of its spontaneous breakup and the suitability of an ADR capture technology to capture an object based on its breakup criticality and level of uncooperativeness [6].

The implementation of the developed domain ontology is performed within the TRACER software library. The development workflow is depicted in Fig. 3 and is divided into two main processes: database generation and software implementation. The former includes collection and pre-processing of raw structured and unstructured data of individuals of the ontology, using the Python programming language. The latter consists instead of activities to implement the domain ontology using the ontology editor Protégé Desktop [19]. The particularities of the TRACER implementation in Protégé consist of: a) minimal hierarchy of classes, b) usage of ontology design pattern, c) differentiation between space debris objects and key characteristics, in terms of individuals [14].

The effectiveness of the developed ontology was tested by applying it onto a database representative objects consisting of 30 large intact cataloged objects (19 payloads and 11 rocket bodies), for which information about their attitude states was available from publicly available resources. The variety of considered ADR capture methods varies from more familiar ones such as manipulator- and net-based to more exotic ones such as electromagnetic- and ablation-based. The obtained results confirm the capability of TRACER to capture, in a standardized, formal and machine-interpretable way, the domain knowledge of payloads and rocket bodies) useful to ADR while at the same time providing tools necessary to transparently handle the input data from an existing database of space debris into an ontology [6].

### 3.2. Using of ontology within astronaut assistance task on lunar environment

The need to use a collection of knowledge from different aspects is a must for robotics as well as for

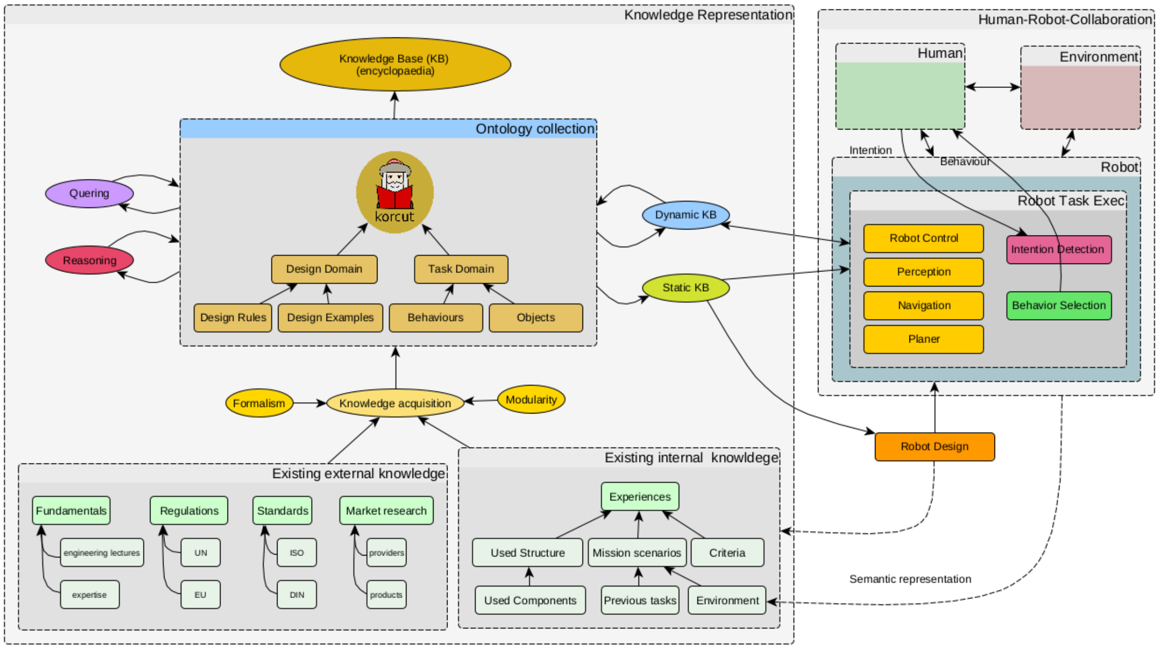

*Figure 2. The overview of the relationship between robotic task and human interaction and the role of knowledge representation.*

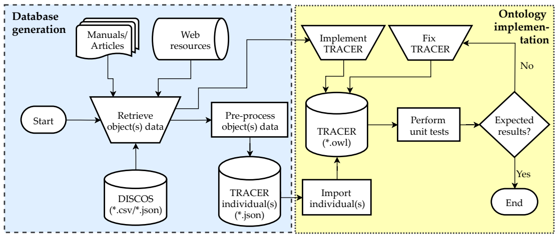

*Figure 3. Development workflow of TRACER software library (refer to the ISO 5807:1985 standard [20] for the definition of symbols) (reprinted from [14]).*

other domains. The goal is to enable the explicit representation as well as the analysis of domain assumptions on a standardized knowledge structure of information and if possible generate further implicit information and keep it reusable and transferable. *Knowledge-based Open Robot voCabulary as Utility Toolkit (korcut)*[21] is a collection of ontologies covering different perspectives of robotics domains. *Korcut* aims to develop a methodology to outline semantic definitions of robot-related objects in different robot domains and provides a description of mathematically-formally well-defined components. *Korcut* covers in general the design and task related domains of terrestrial robotics, which collects robot components, its relation and impacts on system and desired, like design rules, or evaluation criteria with connection to robot requirements, desired configuration and estimated capabilities. We are currently developing a domain ontology named *"korcut ontology to support astronaut for extraterrestrial purposes (kastro)"*, which aims to provide the necessary knowledge for the relationship between an assistance robot and an astronaut for the exemplary daily task on a future lunar site as within the project "Adaptive software framework for context-sensitive, intuitive man-machine-interaction (KiMMI-SF)" at German Center for Artificial Intelligence (DFKI) Robotics Innovation Center (RIC).

Most state-of-the-art man machine interaction (MMI) approaches work really robust as long as they are limited to a specific context. In contrast, KiMMI-SF aims at providing a framework which is capable of detecting the current intention of the human based on the context, as well as detecting possible context changes, e. g., if the human finished working on a task or cancels working on it. To achieve this, the framework requires an expandable knowledge representation to store prior and dynamic knowledge, depending on possible tasks the astronaut needs assistance with. *Kastro*, as part of *korcut*, is a way to support the application of ontology in the high-level robot control process (see in Fig. 2).

The *kastro* in the task domain aims to fill the knowledge gap necessary to identify the astronaut's intent and specify correct desired targets (tools). *Kastro* assists in detecting problems in the environment and environmental objects according to their condition as well as the astronaut's behavior and in finding the desired tool. It provides the requested object with its spatial (physical information of the tool, estimated storage location, etc.) and temporal and time depended information. The demanded knowledge was covered in ontological form, which represents this relationship between the repair task, the astronauts' intentions, and the required tools with its static and dynamic properties. Components of KiMMI-SF are then able to query and update knowledge in *kastro*.

### Development of the kastro ontology

It is important to know the details about the covered domains in order to develop an ontology, focusing on the required information about the elements and properties as well as the use cases. Expert guidance is required to determine the correct domain to work in and to parameterize the elements to define a correct formal description. The ontology development metrics developed by Noy & McGuinness [22] and De Nicola et al.[23] are used to guide the design and implementation steps of the *kastro*. Based on this works, the essential components for building the ontology that shape the ontology are the storyboard, the competency questions, and the vocabulary, as following:

**Story board:** An astronaut's daily exemplar task include observing the station's environment and ensuring that utility systems such as solar and ventilation systems are working properly. In this sense, the systems and components must be monitored telemetrically as well as inspected visually or mechanically on site. In case of anomalies, the problem must be quickly detected and corrected to avoid further hazards. *Autonomous Rover Team for Exploration and Manipulation (Artemis)*[1] is a six-wheel drive autonomous mobile robot (AMR). Each wheel of the *Artemis* has 2 degree of freedom (DoF) and these wheels are distributed in pairs on 3 passive swing arm suspensions around the system. The system is equipped with a *Compliant Robot Arm (Compi)*[2] identical manipulator arm with 6 DoF, on which a two-finger gripper with one active joint and four passive joints is mounted. This robot would assist the astronaut in his tasks and would also be able to perform his subtasks. Finding, bringing, or returning a tool or carrying an object for the astronaut helps minimize the astronaut's resources and risks such as reducing the time to complete the task in a non-atmospheric environment. Thereby, the robot will assist the astronaut during the inspection and recognize his gestures to correctly interpret his commands and bring the required tools or other items, as well as search for the missing objects in the appropriate places if they are temporarily absent, or leave the used objects back to storage.

The demanded knowledge was covered in ontological form, which represents this relationship between the repair task, the astronauts' intentions, and the required tools with its static and dynamic properties.

**Competency questions:** The following competency questions were considered in the development of kastro.

- What objects in the environment can be repaired by astronauts?
- What problem can an object have?
- What methods are useful to fix the problem associated with the object?
- What tools are stored in what specific location?
- Which tool must be used to repair the object?
- Where can this tool be temporarily stored?
- What are the stationary and dynamic properties of a tool?

**Terminology** Within respect to by KiMMI-SF project targeting demo scenario, the vocabulary of main classes is following,

- Environmental object
- Object problem
- Hand tools
- Tool function
- Workshop item

According these KiMMI-SF goals related points the determination of the scope of *kastro* is illustrated as concept map in Fig. 4. Exemplar concept description of a detected object in a test case and the implicit defined class for defect type are expressed in Manchester OWL syntax as follows. The individual `detected_object_1` representing a detected object during astronaut-robot collaboration task. It is automatically generated with properties from dynamic information provided by the KiMMI-SF subtask based on the static information from the *kastro* ontology.

```
detected_object_1
  hasBoundingBoxDimensionX 1,
  hasBoundingBoxDimensionY 1,
  hasBoundingBoxDimensionZ 0,
  hasEnvironmentalObjectType
    object_pipe,
  hasRecognizedDefectType
```

---

[1] https://tinyurl.com/artemis-dfki

[2] https://tinyurl.com/compi-dfki

```
  type_defect_pipe_hole_on
```

The class `Pipe_Hole_Small` is a defined class that obtains its members by reasoning that satisfies the following defined axioms. Corresponding individuals also inherit by reasoning the new relation to `instruction_repair_hole_small`, which is used to further determine the required repair instruction method.

```
Pipe_Hole_Small SubClassOf
  Repair_Introduction
Pipe_Hole_Small EquivalentTo(
  ((hasRecognizedDefectType value
    type_defect_pipe_hole_on)
  and ((hasBoundingBoxDimensionX
    some int[≥ 0, ≤ 2])
  and (hasBoundingBoxDimensionY
    some int[≥ 0 ≤ 2]))),
  (hasRepairInstruction value
    instruction_repair_hole_small))
```

**_KiMMI-SF_ framework in terms of connection to the ontology in the application.**

Within the *KiMMI-SF* framework several Robot Construction Kit (Rock)[3,4][24] tasks are started, depending on the current context. Each task provides measurements of the environment, the human or the robot. These are interpreted by the framework as context variables. Based on these variables and prior knowledge the framework can decide if the current behavior needs to be changed by using a Bayesian approach [25]. To have a unified interface to query and change prior knowledge, *korcut* is accessed via a Python library based on Owlready[26]. This library supports the developer with numerous functions to access the *korcut* ontologies in the form of rules, relations or collection of concepts as metamodels. This can be used to acquire static information, such as information about the kinematic structure of the robot or the causality of the task-related solution approach. In addition, it is also possible to store, reason about, and query dynamic information, such as the remaining battery and resources needed for the dynamic task conditions, the current location of the astronaut, or the tracking of the targeted or requested object or their dynamic properties. *Kastro* is interfaced by using this library with other *KiMMI-SF* modules such as *intention recognition* and *context manager* via the *Rock-Python* module. An example of the process is presented in Algorithm 1.

Through the work, the semantic definition of the repair task as a collaboration between astronaut and robot was focused. With the developed work steps and the implemented ontology as well as an additional library, the application of the ontology for the task domain for human-relevant tasks for robot support was integrated and partially tested on simulation level. The ontology supports both the decision for the necessary repair steps based on the problem state and the knowledge of the location of the necessary tool, which in turn has to be requested and located. The ontology is also prepared for the changes that may occur over time, such as changing the location or state of objects over time.

**Algorithm 1** Knowledge acquisition algorithm in *KiMMI-SF* process

```
Reason the ontology.
If an intention is received about a found fault, query the area based on the localization parameters.
Query the area specific environment objects
Query the possible fault types for this type of object.
Query the detected defect type of the object with specific parameters.
Query appropriate repair instructions to determine the correct repair method, the necessary repair tool with its physical properties and its expected storage with spatial information.
Query the robot's ability and current state to pick up and bring the desired tool
If the robot is able to pick up the tool, then plan the trajectory and drive to bring the tool.
If the tool cannot be found at the estimated position, then query possible alternate positions to verify the requested tool.
If the tool is found at the alternative position, update the parameters.
```

## 4. DISCUSSION

In the context of our work, we encounter the problem of *"information paradox"*, i.e., the lack of coherence of the available information, the missing linking of concepts between disciplines, or the fact that some information is not suitable for common use, defined stored and used internally by several threads several times in different ways. Moreover, we have more and more information at hand, but we *"recognize - understand - apply"* only a part of it, i.e. we are not able to extract useful information from it. Ontology helps to make this association. Furthermore, ontologies allow the process or framework to adapt to different scenarios/requirements with changes over time such as objects becoming obsolete or resources changing and their current state without having to adapt the program logic. In this sense, ontologies are needed also in the space domain and are currently finding their way, albeit at a reduced pace when compared to other domains

---

[3] `https://tinyurl.com/rock-DFKI`

[4] `https://www.rock-robotics.org/`

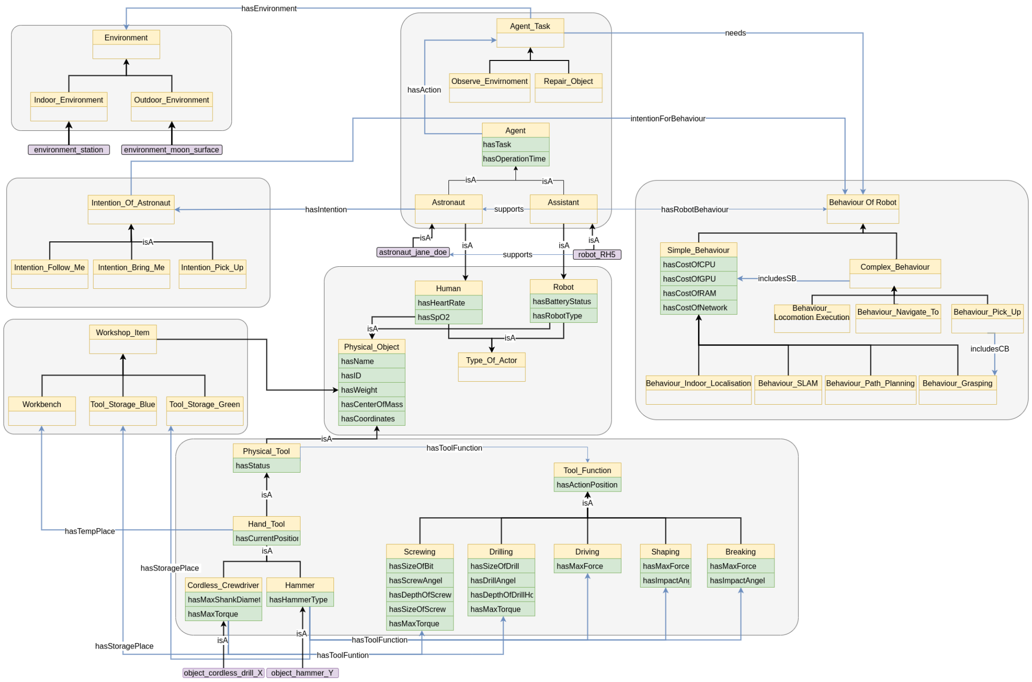

*Figure 4. Concept map of* kastro *ontology that illustrates the concepts and their interrelationships in the astronaut-robot interaction domain.*

such as the Word Wide Web or medicine. To fill this gap (even though only partially), we have developed and presented 2 ontologies in the ADR and planetary exploration domains. TRACER relies on accessible data. With new accessible data in ADR domain, TRACER allows more objects to be included and can be used for analysis of the most ADR capture method(s). On the other hand, kastro has the task-dependent limitation in the current development phase, which will be reduced in the later project phase.

## 5. CONCLUSION

In this paper we provide an overview of practical methods to address the problem with static and/or dynamic knowledge in two realistic application scenarios. TRACER represents a domain-ontology, for data processing, collection, storage and sharing of characteristics of large, intact objects, able to perform automatic ADR capture method(s) selection. *Kastro*, as part of korcut, is a way to support the application of ontology in the high-level robot control process. *Kastro* in the task domain aims to fill the knowledge gap necessary to identify the astronaut's intent and specify correct desired targets (tools). Both ontologies provide domain-specific knowledge for domain internal and external experts or non-experts that supports applications in ADR and MMI tasks. They will be continued in the current and future projects as part of ongoing development for additional interfaces and coverage aspects.

## ACKNOWLEDGEMENTS

The work presented in this paper has been performed in part in the KiMMI-SF project funded by the German Federal Ministry of Economic Affairs and Climate Action (BMWK), and German Aerospace Center e.V. (DLR) under the grant agreement No. 50RA2021.

## REFERENCES

[1] Alberto Olivares-Alarcos, Daniel Beßler, Alaa Khamis, Paulo Goncalves, Maki K. Habib, Julita Bermejo-Alonso, Marcos Barreto, Mohammed Diab, Jan Rosell, João Quintas, and et al. A review and comparison of ontology-based approaches to robot autonomy. *The Knowledge Engineering Review*, 34:e29, 2019.

[2] Maria A. Cornejo-Lupa, Yudith Cardinale, Regina Ticona-Herrera, Dennis Barrios-Aranibar, Manoel Andrade, and Jose Diaz-Amado. Ontoslam: An ontology for representing location and simultaneous mapping information for autonomous robots. *Robotics*, 10(4), 2021.

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

[4] Sumaira Manzoor, Yuri Goncalves Rocha, Sung-Hyeon Joo, Sang-Hyeon Bae, Eun-Jin Kim, Kyeong-Jin Joo, and Tae-Yong Kuc. Ontology-based knowledge representation in robotic systems: A survey oriented toward applications. *Applied Sciences*, 11(10), 2021.

[5] Craig Schlenoff, Anthony Pietromartire, Zeid Kootbally, Stephen Balakirsky, and Sebti Foufou. Ontology-based state representation for intention recognition in cooperative human-robot environments. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 810–817, New York, NY, USA, 2012. Association for Computing Machinery.

[6] Marko Jankovic, Mehmed Yüksel, Mohammad Mohammadzadeh Babr, Francesca Letizia, and Vitali Braun. Space debris ontology for adr capture methods selection. *Acta Astronautica*, 173:56 – 68, 2020. `http://www.sciencedirect.com/science/article/pii/S0094576520301752`,Accessed: 2021-03-22.

[7] ESA Space Debris Office. Database and Information System Characterising Objects in Space (DISCOS).

[8] SAIC. Space-Track.Org.

[9] Alexander P. Cox, Christopher K. Nebelecky, Ronald Rudnicki, William A. Tagliaferri, John L. Crassidis, and Barry Smith. The Space Object Ontology. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 146–153. IEEE.

[10] Robert J. Rovetto. An ontological architecture for orbital debris data. 9(1):67–82.

[11] R. Furfaro, R. Linares, D. Gaylor, M. Jah, and R. Walls. Resident Space Object Characterization and Behavior Understanding via Machine Learning and Ontology-based Bayesian Networks. pages 1–14. Maui Economic Development Board, Inc.

[12] Bin Liu, Li Yao, and Dapeng Han. Harnessing ontology and machine learning for RSO classification. 5(1).

[13] Samantha Le May, Brett Carter, Steve Gehly, and Sven Flegel. Leveraging Web data and graph structures to support rapid space object identification. In *69th International Astronautical Congress (IAC)*, pages 1–10. International Astronautical Federation (IAF).

[14] Marko Jankovic. *Characterization and Capture of Space Debris Objects Using Domain Ontology and Optimal Control*. PhD thesis, Universität Bremen, Bremen, Germany, March 2022.

[15] Mehmed Yüksel, Manuel Vaca Benitez, Luis, Dinmukhamed D. Zardykhan, and Frank Kirchner. Mechatronical design and analysis of a modular developed exoskeleton for rehabilitation purposes. In *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 711–716, 2017.

[16] C.M. Keet. *An Introduction to Ontology Engineering*. 2020.

[17] Robert Arp, Barry Smith, and Andrew D. Spear. *Building Ontologies with Basic Formal Ontology*. 2015.

[18] V. Agrapov, F. Alby, C. Bonnal, C. Cazaux, E. Christiansen, J.-C. Dolado-Perze, D. Finkelman, S. Kibe, H. Klinkrad, H. Krag, P. Krisko, J.-C. Liou, D. McKnight, T. Masson-Zwaan, C. Mathieu, M. Metz, D. Oltrogge, F. Schaefer, T. Schlidknecht, and M. Sorge. IAA Situation Report on Space Debris - 2016.

[19] Mark A. Musen. The protégé project: A look back and a look forward. *AI Matters*, 1(4):4–12, June 2015. `https://doi.org/10.1145/2757001.2757003`, Accessed: 2021-03-22.

[20] ISO IEC JTC 1/SC 7. ISO 5807:1985 - Information processing– Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.

[21] Mehmed Yüksel. Korcut ontology family on Zenodo, 2021. `https://doi.org/10.5281/zenodo.4457870`, Accessed: 2021-03-20.

[22] Natalya F. Noy and Deborah L. Mcguinness. Ontology development 101: A guide to creating your first ontology. Technical report, 2001.

[23] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. A software engineering approach to ontology building, 2009.

[24] Sylvain Joyeux, Jakob Schwendner, and Thomas M. Roehr. Modular Software for an Autonomous Space Rover. In *12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2014)*, Montreal, Canada, 2014.

[25] Adrian Lubitz, Octavio Arriaga, Teena Hassan, Nina Hoyer, and Elsa Andrea Kirchner. A bayesian approach to context-based recognition of human intention for context-adaptive robot assistance in space. In *SpaceCHI 2.0, Advancing Human-Computer Interaction for Space Exploration*.

[26] Jean-Baptiste Lamy. Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial Intelligence in Medicine*, 80:11–28, 2017. `https://www.sciencedirect.com/science/article/pii/S0933365717300271`,Accessed: 2021-03-22.