Ontological Framework to Improve Motion Planning of Manipulative Agents through Semantic Knowledge-based Reasoning

Rodrigo Bernardo[1,2,*], João M. C. Sousa[1] and Paulo J. S. Gonçalves[2]

[1] *IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Portugal*
[2] *IDMEC, Instituto Politécnico de Castelo Branco, Castelo Branco, Portugal*

Abstract

This paper describes the actions taken in developing a framework that aims to improve the motion planning of a manipulative robotic agent through reasoning based on semantic knowledge. The Semantic Web Rule Language (SWRL) was employed to draw new insights from the existing information about the robotic system and its environment. Recent ontology-based standards have been developed (IEEE 1872-2015; IEEE 1872.2-2021; IEEE 7007-2021), and others are currently under development (IEEE P1872.1; IEEE P1872.3) to improve robot performance in task execution. Ontological knowledge “semantic map" was generated using a deep neural network trained to detect and classify objects in the environment where the manipulator agent acts. Manipulation constraints were deduced, and the environment corresponding to the agent's manipulation workspace was created so the planner could interpret it to generate a collision-free path. Several SPARQL queries were used to explore the semantic map and allow ontological reasoning. The proposed framework was implemented and validated in a real experimental setting, using the ROSPlan planning framework to perform the planning tasks. This ontology-based framework proved to be a promising strategy. E.g., it allows the robotic manipulative agent to interact with objects, e.g., to choose a mobile phone or a water bottle, using semantic information from the environment to solve the requested tasks.

Keywords

Knowledge representation, Ontologies, Manipulation, Motion planning, Semantic maps

1. Introduction

Within the ongoing advances of Industry 5.0, robotic systems are increasingly present in highly dynamic environments, including environments shared with humans [1]. The description of a Cyber-Physical Systems, like a Human-Robot Collaborative scenario, requires a model of complex adaptive behaviors of involved agents from both a “local perspective" (i.e., the point of view of an agent) and a “global perspective" (i.e., the point of view of the production and related constraints and objectives) [2]. The need to find efficient paths (motion planning) for manipulator robots, and new effective strategies to manipulate different objects to perform more complex tasks, is crucial for various real-world applications.

RobOntics 2023: Workshop on Ontologies in Autonomous Robotics, August 28, 2023, Seoul, South Korea
*Corresponding author.
rodrigo.f.bernardo@tecnico.ulisboa.pt (R. Bernardo); jmsousa@tecnico.ulisboa.pt (João M. C. Sousa); paulo.goncalves@ipcb.pt (Paulo J. S. Gonçalves)

Traditionally, there are two approaches to the problem: offline planning, which assumes a perfectly-known and stable environment, and online planning, which focuses on dealing with environmental uncertainties. The majority of motion planning algorithms are based on random sampling algorithms [3, 4, 5, 6, 7]. Recently, algorithms such as optimization-based [8], Probabilistic Movement Primitives (ProMPs)-based [9, 10] and physics-based methods [11, 12], have been shown to be more effective [13]. However, these methods used in trajectory planning become limited if the actions required to perform the task are subject to strong geometric constraints of the environment (lack of space to place objects, occlusions) and of the robot (accessibility of objects, kinematic constraints of the manipulators) [14]. Ontologies have shown great potential in improving the motion planning of agents in symbiotic work systems (e.g. a team of robots can work together to assemble a product on the shop floor, each robot being responsible for a specific task, without them clashing) [15, 16, 17].

The main goal of this paper is to develop a framework where motion planning is improved, allowing the possibility to reconfigure the initially defined plan (e.g. recovery from a situation where an unexpected obstacle appears on the path) for a robotics manipulative agent through semantic knowledge and reasoning (about manipulation actions and the objects present in the environment). A deep neural network was trained to detect and classify objects in the environment where the robotic agent is located. This information created a semantic map (described in an ontology) of the environment. The semantic maps deal with metainformation that models the properties and relationships of relevant concepts in the domain encoded in a Knowledge Base (KB). Semantic maps enable the execution of high-level robotic tasks efficiently, and several strategies have been presented [18, 19]. The proposed domain ontology is based on the recently developed ontology-based standards (IEEE 1872-2015; IEEE 1872.2-2021; IEEE 7007-2021); others are currently under development (IEEE P1872.1; IEEE P1872.3). The proposed domain ontology has as a foundational layer the top-level ontology DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) to follow the interpretation of some relevant concepts and thus define an ontology (domain) with a high level of flexibility [20]. The Semantic Web Rule Language (SWRL) [21] was employed to draw new insights from the existing information about the robotic system and its environment. Several SPARQL queries were used to explore the semantic map and allow ontological reasoning [22].

Several works have already presented strategies using ontologies to improve motion planning in dynamic environments [23, 24, 25, 26]. However, these works present isolated approaches that do not use an upper ontology as a base, making them difficult to reusable. The proposed ontology intends to serve as a basis for future works which focus on improving motion planning because, to date, existing approaches have not presented inference systems to increase the robot's semantic knowledge of the environment and the new objects that can arrive at the environment, in order to improve motion planning.

Experimental validation is performed in a simple house environment based on a smart-home environment. Knowledge was inferred based on semantic knowledge, and the ROSPlan[1] framework was used to perform task planning based on the actions defined in the ontology.

The following section presents the robotic and semantic knowledge-based reasoning developed to improve the motion planning of robotic manipulator agents in non-deterministic environments.

[1] https://kcl-planning.github.io/ROSPlan/

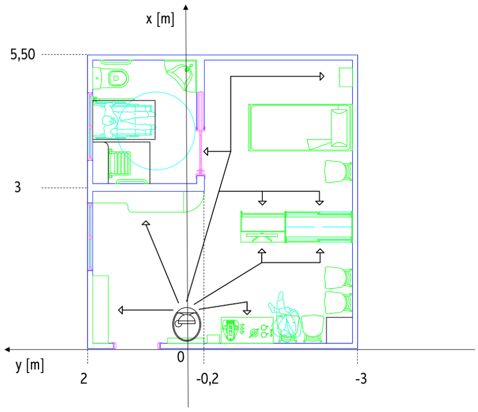
Section 3 presents the conceptual framework of the global knowledge engine and an example of a practical validation of the proposed framework in a real environment. The paper concludes with conclusions and future work.

2. Robot and Semantic Knowledge-based Reasoning

2.1. Robot and Smart Home Description

Figure 1a illustrates the Autonomous Mobile Manipulator Robot (AMMR), used in the study. The mobile platform holds a Universal Robot UR3[2] robotic manipulator. The robot manipulator is equipped with a Robotiq 2f-140[3] gripper, designed to handle objects. The mobile base is also equipped with various sensors, including rear and front lasers, video cameras, and RGB-D camera. These sensors enable autonomous navigation, obstacle detection, and recognition, allowing the robot manipulator to perceive its working environment. The experimental validation is realized in a simple house environment based on a smart-home environment built in the robotics laboratory of Instituto Politécnico de Castelo Branco (Fig. 1b).

(a) AMMR.



(b) Metric representation of the rooms.

Figure 1: Autonomous Mobile Manipulator Robot (AMMR) used in experimental validation and a Smart-home environment [27].

2.2. Semantic Knowledge-based Reasoning

Recent ontology-based standards were developed (IEEE 1872-2015; IEEE 1872.2-2021; IEEE 7007-2021), and others are currently in development (IEEE P1872.1; IEEE P1872.3) to improve robot performance while executing tasks. This is a very hot topic in current standardization efforts worldwide. This section presents the first efforts to integrate the concepts defined in the standards mentioned above, which has as a foundational layer the upper ontology DOLCE to accompany

[2]https://www.universal-robots.com/pt/produtos/ur3-robot/
[3]https://robotiq.com/products/2f85-140-adaptive-robot-gripper

the interpretation of some relevant concepts and thus builds a prototype ontology with a high level of flexibility for the specific domain.

Figure 2: Snapshot of the ontology.

The ontology was created using the Protégé software, version 5.6.1 [28]. To ensure the consistency of the ontology and inferring reasoning based on it, the Pellet logic Reasoner version 2.2.0 was used [29]. To access the developed OWL ontology, the Owlready2 library was utilized. Owlready2 is a Python module that allows loading, modification, saving, and reasoning with OWL 2.0 ontologies [30]. Following the above-stated assumptions, figure 2 presents the hierarchical class where the main concepts are defined in the proposed domain ontology. It was designed for an agent to interpret and interact with its surrounding environment.

Several Object properties were created (e.g., `hasCapability`, `part_of`, `Located_at`, etc.), to relate the concepts in ontology so that an agent can characterize its environment and Data properties (e.g., `position_x`, `Reach`, `state_voltage`, `BatteryLevel`, etc.) for providing relation to attaching an entity instance to some literal datatype value. Different concepts were added to the ontology to confer modularity. In order to be possible to infer knowledge such as the possible paths that a robot can take (`FreePath`; `CollidingPath`) and the possible trajectories (`FreeTrajectory`; `CollidingTrajectory`). The concept `TaskState`, correlates the state of different tasks: For example: `ObjectState`: `FixedObject` and `ManipulableObject` in the environment to correctly identify if an object is manipulable (i.e. if it is within the manipulator's workspace); the different actions that agents can perform based on the agent type (e.g. `Mobile_actions`, `Manipulator_actions`, `Gripper_actions`); etc.

SWRL are used to infer new knowledge. For example: based on the knowledge of the dimensions of the rooms of the environment (Fig. 1b), the following SWRL rule (1) is used to identify the space of the environment in which the agent is (e.g., the rule to check if the agent is in the `LivingRoom_1`).

$$
\begin{aligned}
&Agent(?Ag) \wedge position\_x(?Ag, ?px) \wedge position\_y(?Ag, ?py) \wedge \\
&\quad swrlb : greaterThanOrEqual(?px, 0) \wedge swrlb : lessThanOrEqual(?px, 3) \wedge \\
&\quad swrlb : greaterThanOrEqual(?py, -3) \wedge swrlb : lessThanOrEqual(?py, 2) \\
&\qquad \rightarrow located\_at(?Ag, LivingRoom\_1)
\end{aligned} \tag{1}
$$

Throughout the plan execution, the properties of the specific instance of the manipulator (`ur3_arm`) are continuously monitored and represented. These properties, known as data properties, are dynamically updated and stored in the framework's database. This enables the overall system to have real-time information about the current state and status of the manipulator. By constantly updating and storing the data properties, the framework ensures that the global system is aware of any changes or updates in the state of the manipulator robot, facilitating effective coordination and decision-making during the execution of the plan.

Considering the information about the distance of an object at the manipulator agent and the reach of the manipulator, we can formulate an SWRL rule using the concepts from the ontology to automatically determine the objects present in the workspace, specifically in the `ManipulationRegion` (See SWRL (2)).

$$
\begin{aligned}
&AllObjects(?obj) \wedge robotic\_arm(?r) \wedge Reach(?r, ?re) \\
&\quad \wedge EuclideanDistance(?obj, ?dist) \wedge swrlb : lessThan(?dist, ?re) \\
&\qquad \rightarrow located\_at(?Obj, ManipulationRegion)
\end{aligned} \tag{2}
$$

3. Global Knowledge Engine Conceptual Framework

The proposed framework comprises different main modules. The "Knowledge-based reasoning engine (Ontology)" has previously been described. The module "Hardware-level of robotic agent (Robotic manipulator)" is composed of two toolboxes, one for the robotic manipulator UR3 (toolbox_ur3) and another for the gripper 2f-140 from Robotiq (toolbox_gripper), these were developed to work on the ROS. They are written based on the ur-rtde [4] library, which communicates with the UR3 via the real-time data exchange protocol (RTDE). The "Perception (object detection)" module uses a deep neural network to detect and classify objects in the environment where the robotic agent is. Based on this information, a semantic map was created. The YOLO v3 object detection algorithm [31] from Darknet for ROS was used for object detection [32]. The main module of the framework is titled "Task Manager (Behavior tree)", it is composed of a Behavior Tree (BT), which is designed to perform all the management of the framework presented in this paper. BT is a graphical representation of the control logic for autonomous agents. This module is also composed of all the architecture used in the trajectory planning of the manipulator; the Probabilistic Roadmap Method (PRM) algorithm was used, which was applied based on the Robotics Library (RL) [33]. Semantic knowledge was used to improve the motion planning of the manipulator agent. The ontology is queried to identify all objects

[4] https://sdurobotics.gitlab.io/ur_rtde

and their properties (e.g. dimensions, weight, position on the map, etc.) in the robot's working area. Based on the semantic information, the scenario around the manipulator agent is created to create a collision-free path and replan a new path if a change in the environment makes the initial path unfeasible. The ROSPlan [34] framework was used to perform the planning tasks in this framework based on a PDDL (Planning Domain Definition Language) problem and domain. It is a high-level tool proven well for planning in the ROS environment. The PDDL problem file is then defined using the instances of the ontology for the specific task needed to be performed by the robot. The information for each one of the instances is stored using a MongoDB database. Different action interfaces have been written to control the AMMR (Fig. 1a), i.e., its `Mobile agent actions`, `Manipulator agent actions`, and `Gripper actions` for proper interaction with the robot presented previously. These interfaces are constantly listening to the action ROSPlan messages. Moreover, the MongoDB database was used for semantic memory storage, e.g., fixed locations, robots, objects and their properties, goal parameters, etc.

All the software developed to control the robot is developed using Robot Operating System (ROS) Noetic and Ubuntu 20.04.4 LTS operating system with an Intel® Core™ i7-7740X CPU @ 3.30 GHz $\times$ 8 processor, 16 GB RAM, and Quadro P2000/PCIe/SSE2 Graphics. The input video is obtained using Intel® RealSense™ D415i Depth Camera.

3.1. Validation of the Proposed Framework in a Real Environment

An experiment was conducted to validate the proposed framework using the experimental setup shown in Figure 3a. The goal of the experiment was to pick an object called `cell_phone_1`, which had been previously identified (as seen in Figure 3b). The information about this object and the robot's location in the living room was previously stored in the knowledge base. To create a challenging scenario, an object named `bottle_1` was deliberately placed in front of the target object, making it impossible to pick up the `cell_phone_1` without colliding with the obstructing object. This obstruction also made the `cell_phone_1` object unobservable, as shown in Figure 3c.

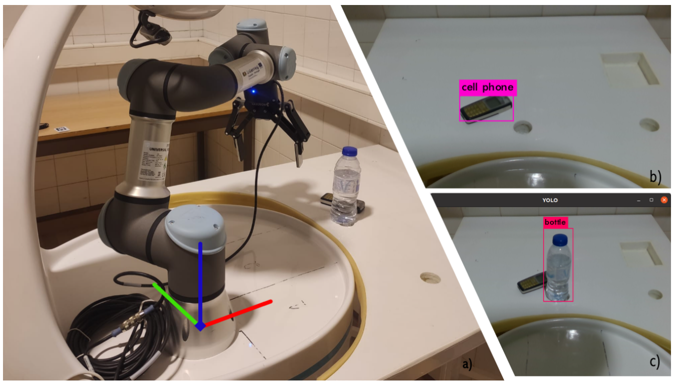


Figure 3: Pratical evaluation setup. a) Apparatus overview. b) First observation of the environment. c) Final observation of the environment, where the cell phone is no longer observed, as a bottle was placed covering it [27].

A simple pick and place plan can be generated in the scenario depicted in Figure 3b. The initial plan is shown in Figure 4a. However, in the scenario shown in Figure 3c, the initial plan fails because the position of the `bottle_1` object obstructs the picking movement, rendering the initial plan invalid. The ontology is queried to determine if the obstructing object; i.e., the object is `located_at` in `ManipulationRegion` of manipulator agent based on SWRL (2). Can be manipulated based on the manipulator and gripper characteristics (as shown in listing 1). The query confirms the object is manipulable (Fig. 5).

(a) Generated initial plan. (b) Generated final plan.

Figure 4: Generated PDDL plans [27].

Based on this information, a new problem is formulated in PDDL to generate a plan that allows the successful retrieval of the `cell_phone_1` object without any collision (as shown in Figure 4b). The revised plan consists of three sets of tasks: `perceive_object`, `pick`, and `drop`. The first task involves moving the obstacle (`bottle_1`) to a known position at the base of the mobile agent, which is stored in the knowledge base (e.g., `right_platform`). The second task is to pick up the `cell_phone_1` and place it at another known position in the mobile agent's base (e.g., `left_platform`). Finally, the last task involves returning the `bottle_1` object to its initial position where it was initially observed.

Listing 1: Question. Are the objects present manipulable?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX on: <http://www.semanticweb.org/idmind_ur3#>
SELECT ?arm ?obj ?obj_dim_x ?obj_dim_y
WHERE { ?arm rdf:type on:robotic_arm. ?arm on:hasCapability ?tool.
        ?arm rdf:type on:CandidateForMission.
        ?tool on:Grasping ?tool_grasping.
        ?obj rdf:type on:ManipulationRegion.


    ?obj on:box_xmin ?obj_box_xmin. ?obj on:box_xmax ?obj_box_xmax.
    ?obj on:box_ymin ?obj_box_ymin. ?obj on:box_ymax ?obj_box_ymax.

    BIND((?obj_box_xmax - ?obj_box_xmin) AS ?obj_dim_x)
    BIND((?obj_box_ymax - ?obj_box_ymin) AS ?obj_dim_y)

BIND(IF((obj_dim_x<?tool_grasping)||(obj_dim_y<?tool_grasping),
                                        "TRUE","FALSE") AS ?manipulable)
```

?obj	?arm	?tool	?manipulable
on:bottle_1	on:ur3_arm	on:gripper_2f_140	TRUE^^xsd:string
on:cell_phone_1	on:ur3_arm	on:gripper_2f_140	TRUE^^xsd:string

2 results

Figure 5: Response of query listing in 1.

By re-planning the movement, the initial objective is successfully achieved. Although the newly generated plan takes 67.52% more time compared to the initial plan, it eliminates the need for manual intervention to remove the obstacle, which would have resulted in higher costs. It should be noted that the presented framework has a limitation in that it only works if the objects causing the collision are manipulable. If an unmanipulable object were responsible for the collision, the system would send an error message indicating that the goal cannot be reached, and a solution would require changing the position of the AMMR base to a location that allows the retrieval of the `cell_phone_1`. However, implementing such behavior using the mobile base capability is beyond the scope of this work and will be a focus of future research.

4. Conclusions and Future Work

This paper presented an ontological framework to improve the motion planning process, giving the possibility to reconfigure the initially defined trajectory (e.g. recovery from a situation where an unexpected obstacle appears on the path) for a robotic manipulator agent. A deep neural network trained to detect and classify objects in the environment where the robotic agent acts where used to create a semantic map of the environment (using some concepts from IEEE 1872.2-2021 standard). The semantic map and SWRL rules were used to infer new knowledge based on the known environment and the robotic system. Several SPARQL queries were used to explore the semantic map and allow ontological reasoning. The proposed framework was implemented in a real scenario using the ROSPlan, and its potential was proven through a real manipulation situation. Efforts are underway to complete the reasoning framework; it is imperative to develop solutions that exploit the capabilities of an AMMR agent, for example, that optimise manipulation tasks based on the movement of the mobile base of the robotic agent, etc.

Acknowledgment

This work is financed by national funds through FCT - Foundation for Science and Technology, I.P., through IDMEC, under LAETA, project UIDB/50022/2020. The work of Rodrigo Bernardo was supported by the PhD Scholarship BD/6841/2020 from FCT. This work has received funding from: the project 0770_EUROAGE2_4_E (POCTEP Programa Interreg V-A Spain-Portugal), and the European Union's Horizon 2020 programme under StandICT.eu 2023 (under Grant Agreement No.: 951972).

References

[1] S. Nahavandi, Industry 5.0—a human-centric solution, Sustainability 11 (2019) 4371.

[2] S. Borgo, An ontological view of components and interactions in behaviorally adaptive systems, Journal of Integrated Design and Process Science 23 (2019) 17–35.

[3] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, The international journal of robotics research 30 (2011) 846–894.

[4] T. Siméon, J.-P. Laumond, J. Cortés, A. Sahbani, Manipulation planning with probabilistic roadmaps, The International Journal of Robotics Research 23 (2004) 729–746.

[5] D. Hsu, J.-C. Latombe, H. Kurniawati, On the probabilistic foundations of probabilistic roadmap planning, The International Journal of Robotics Research 25 (2006) 627–643.

[6] S. M. LaValle, et al., Rapidly-exploring random trees: A new tool for path planning (1998).

[7] S. Rodriguez, X. Tang, J.-M. Lien, N. M. Amato, An obstacle-based rapidly-exploring random tree, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, 2006, pp. 895–900.

[8] O. Salzman, D. Halperin, Asymptotically near-optimal rrt for fast, high-quality motion planning, IEEE Transactions on Robotics 32 (2016) 473–483.

[9] A. Paraschos, C. Daniel, J. Peters, G. Neumann, Using probabilistic movement primitives in robotics, Autonomous Robots 42 (2018) 529–551.

[10] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, J. Peters, Adaptation and robust learning of probabilistic movement primitives, IEEE Transactions on Robotics 36 (2020) 366–379.

[11] N. Kitaev, I. Mordatch, S. Patil, P. Abbeel, Physics-based trajectory optimization for grasping in cluttered environments, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 3102–3109.

[12] M. Moll, L. Kavraki, J. Rosell, et al., Randomized physics-based motion planning for grasping in cluttered and uncertain environments, IEEE Robotics and Automation Letters 3 (2017) 712–719.

[13] S. Liu, P. Liu, A review of motion planning algorithms for robotic arm systems, RiTA 2020 (2021) 56–66.

[14] C. Li, G. Tian, Transferring the semantic constraints in human manipulation behaviors to robots, Applied Intelligence 50 (2020) 1711–1724.

[15] C. Schou, R. S. Andersen, D. Chrysostomou, S. Bøgh, O. Madsen, Skill-based instruction of collaborative robots in industrial settings, Robotics and Computer-Integrated Manufacturing 53 (2018) 72–80.

[16] A. Akbari, Muhayyuddin, J. Rosell, Knowledge-oriented task and motion planning for multiple mobile robots, Journal of Experimental & Theoretical Artificial Intelligence 31 (2019) 137–162.

[17] R. Bernardo, J. M. Sousa, P. J. Gonçalves, Survey on robotic systems for internal logistics, Journal of Manufacturing Systems 65 (2022) 339–350.

[18] R. Bernardo, J. Sousa, P. J. Gonçalves, Planning robotic agent actions using semantic knowledge for a home environment, Intelligence & Robotics 1 (2021) 116–130.

[19] A. Achour, H. Al-Assaad, Y. Dupuis, M. El Zaher, Collaborative mobile robotics for semantic mapping: A survey, Applied Sciences 12 (2022) 10316.

[20] S. Borgo, C. Masolo, Ontological foundations of dolce, in: Theory and applications of

ontology: Computer applications, Springer, 2010, pp. 279–295.
[21] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al., Swrl: A semantic web rule language combining owl and ruleml, W3C Member submission 21 (2004) 1–31.
[22] E. Sirin, B. Parsia, et al., Sparql-dl: Sparql query for owl-dl., in: OWLED, volume 258, 2007.
[23] S. Feyzabadi, S. Carpin, Knowledge and data representation for motion planning in dynamic environments, in: Robot Intelligence Technology and Applications 2, Springer, 2014, pp. 233–240.
[24] A. Akbari, J. Rosell, et al., Ontological physics-based motion planning for manipulation, in: 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), IEEE, 2015, pp. 1–7.
[25] M. Beetz, L. Mösenlechner, M. Tenorth, Cram—a cognitive robot abstract machine for everyday manipulation in human environments, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 1012–1017.
[26] M. Diab, A. Akbari, J. Rosell, et al., An ontology framework for physics-based manipulation planning, in: Iberian Robotics conference, Springer, 2017, pp. 452–464.
[27] R. Bernardo, J. M. Sousa, P. J. Gonçalves, A novel framework to improve motion planning of robotic systems through semantic knowledge-based reasoning, Computers & Industrial Engineering 182 (2023) 109345. URL: https://www.sciencedirect.com/science/article/pii/S0360835223003698. doi:https://doi.org/10.1016/j.cie.2023.109345.
[28] Protégé, Protégé, accessed June, 2022. URL: https://protege.stanford.edu.
[29] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical owl-dl reasoner, Journal of Web Semantics 5 (2007) 51–53.
[30] J.-B. Lamy, Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies, Artificial intelligence in medicine 80 (2017) 11–28.
[31] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv (2018).
[32] M. Bjelonic, YOLO ROS: Real-time object detection for ROS, https://github.com/leggedrobotics/darknet_ros, 2016–2018.
[33] M. Rickert, A. Gaschler, Robotics Library: An object-oriented approach to robot applications, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 733–740. doi:10.1109/IROS.2017.8202232.
[34] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtos, M. Carreras, Rosplan: Planning in the robot operating system, in: Proceedings of the International Conference on Automated Planning and Scheduling, volume 25, 2015, pp. 333–341.