



**HAL**  
open science

# Towards a Knowledge-Based Design Methodology for Modular Robotic System

Lucas Jimenez, Frédéric Demoly, Sihao Deng, Samuel Gomes

► **To cite this version:**

Lucas Jimenez, Frédéric Demoly, Sihao Deng, Samuel Gomes. Towards a Knowledge-Based Design Methodology for Modular Robotic System. 17th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2020, Rapperswil, Switzerland. pp.50-58, 10.1007/978-3-030-62807-9\_5. hal-03753131

**HAL Id: hal-03753131**

**<https://inria.hal.science/hal-03753131>**

Submitted on 17 Aug 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# Towards A Knowledge-Based Design Methodology For Modular Robotic System

Lucas Jimenez<sup>1,2,\*</sup>, Frédéric Demoly<sup>1</sup>, Sihao Deng<sup>1</sup> and Samuel Gomes<sup>1</sup>

<sup>1</sup> ICB UMR 6303 CNRS, Univ. Bourgogne Franche-Comté, UTBM, Belfort, France

<sup>2</sup> MS-INNOV company, Belfort, France

lucas.jimenez@utbm.fr, frederic.demoly@utbm.fr, sihao.deng@utbm.fr, samuel.gomes@utbm.fr

\*Corresponding author

## Abstract.

A particular challenging task in the context of the 4<sup>th</sup> Industrial Revolution consists in increasing production system automation in European small and medium enterprises (SMEs). However, automation – often addressed with robotization – represents major investments that are not always accessible for SMEs, in which small batch production requires change in an efficient manner. In fact, software programming and hardware (in terms of reachability and maximum payload) are the flexibility barrier of current industrial robots. On the other hand, researchers and industry are currently pushing their efforts to develop modular robots (in terms of hardware and software), which are task-specific computable and changeable. In this context, this paper aims to develop a new design methodology combining both knowledge representation for task definition and computational tools for structural and logical design using modular units. To demonstrate its suitability in an industrial context requiring reconfiguration capabilities, specific developments have been made and tested with a simple case.

**Keywords:** Modular robotic; Knowledge-based design; Robotic task planning.

## 1 Introduction

Robotic systems are part of the 4<sup>th</sup> Industrial Revolution and might change the way on which products are designed and manufactured. However, current industrial robots suffer from a lack of reconfiguration capabilities. Actually, they are built for large-scale production and are not suitable enough for small batches with rapid production change as needed in small and medium enterprises (SMEs) [1,2]. Manufacturing SMEs are indeed dynamic organizations and tend to be more responsive and automated [3]. Past academic and industrial efforts have been made towards the development and integration of more flexible robotic systems, enabling fast physical reconfigurations according to manufacturing strategy changes. Thereby such systems still need additional efforts for easier design, integration, and task programming [4].

Beyond current existing solutions in industry, modular robotics seems to be a promising strategy to address these aforementioned challenges. Over the last two decades, this research field has received huge attention from researchers, especially for developing devices able to evolve in hostile environment. However, their technology readiness levels currently are not enough aligned with industrial requirements [5]. Now modular robotic systems provide more versatility, robustness and are cheapest compared to industrial robots [6]. To ensure an industrial adoption, it is important to develop methods and tools for supporting hardware and software (control system) design [6-8]. Although research works have already been focused on reconfiguration and self-reconfiguration strategies in order to adapt a system to a new task or a new manufacturing environment using different computational procedures [6,9], efforts still remain on knowledge representation for robotic system design [4].

Therefore, the main research issue consists in capturing and reusing robot task planning knowledge enhanced by computational capabilities for modular robotic design. Actually, developing a knowledge base with reasoning capabilities will facilitate modular robot structure and behaviour generation over embodiment and detailed design stages, especially in SMEs where resources and robotic knowledge are often limited and partial. This structure of the paper is as follows. Section 2 presents a brief literature review on task-specific robot system design and knowledge processing for robotic task definition where motivation is highlighted. Then, in Section 3, a knowledge-based methodology for modular robotic design is proposed. Lastly, a case study is introduced with a developed computational layer to illustrate the proposal.

## **2 Literature Review**

This section reports significant research works on task-specific robotic system design and knowledge processing for robotic task definition, on which research motivation has been highlighted.

### **2.1 Task Specific Robotic System Design**

When a new manufacturing instruction is defined, required payload and reachability might change. Using modular elements could then be useful to structurally adapt a robotic system to any task. As modular robots are composed of individual parts, many structural combinations are possible. Selecting the right structure fulfilling a given task is thus a challenging issue. In such a context, Desai *et al.* [13] have developed a tool dedicated to the generation of a heterogenous modular robot structure based on the desired path with a modular robot library [13]. The developed tool uses an A\* search algorithm on a weighted tree representing array of structural configurations but does not consider functional specifications (only trajectories). Similarly, Valente [14] has used a stochastic algorithm to support decision during the modular robotic design process. Based on 15 key performance indicators (KPIs) and more than 50 parameters, the developed tool supports robot architecture definition by enabling the robotics engineer to select modules fulfilling multiple tasks. Although this approach seems to be mature,

it lacks programming function and is not suitable for a non-robotic expert. Research efforts become strategic on knowledge representation over the robotic cells design stages to overcome combinatorial complexity of robot structure and robot task planning generation.

## 2.2 Knowledge Processing For Robotic Design And Task Definition

Knowledge-based design is a research domain that has been widely investigated over the last decades [15]. Such research works – covering knowledge acquisition, representation and reuse – have decreased the required expertise with automation procedures while increasing tasks abstractions level. As an example, a knowledge base describing relationships between robot capabilities, robot structure and robot environment associated to a reasoner can infer new knowledge from a partially defined input problem [16]. From this perspective, ontologies have been developed to capture knowledge related to robot design, robot behaviours in terms of controller programming and trajectories planning. Ontologies are a set of concepts, parameters and relationship belonging to the same domain. Ontologies, among others, have been widely used in Knowledge Base System (KBS), for semantic reasoning in engineering design [18]. If appropriately combined, both ontologies provide assistance in robot architectural design for a specific task. Moreover, efforts have also been made for high abstraction task formalization through the KnowRob framework [17]. It can be seen as an ontology-based framework enabling any robotic system to perform tasks based on abstract instructions.

## 2.3 Research Issues

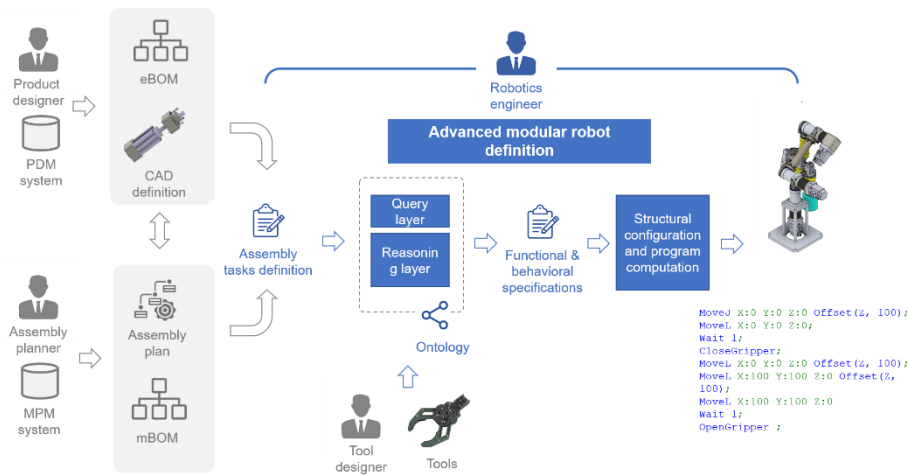
The above-mentioned studies have stressed the importance of both designing and programming modular robots for a specific task. Among the research works, two relevant approaches have been highlighted to automatically generate robot architecture and to support robots design and programming via a knowledge base associated to an inference engine. At first glance, these approaches seem to partially tackle fast reconfiguration planning issues in SMEs. The novelty here consists in combining computational tool and multi-perspective knowledge therefore enabling architectural and programming reconfiguration of modular robots.

## 3 Proposed Methodology

This section gives an overview of the proposed methodology for designing and programming modular robot system based on functional and behavioral specifications, such as pay load or needed tools (functional) and speed or torque (behavioral). In such a routine robotic design context, the proposed framework – illustrated in Fig. 1 – starts by considering the need to implement one or several assembly operations (i.e. manufacturing bill of materials mBOM) defined by the assembly planner in the manufacturing process management (MPM) system. Such information is defined according to the assembly plan that is co-generated with the CAD model of the product and its related

engineering BOM (eBOM) stored in the product data management system (PDM) [19,20]. Among the defined assembly plan, assembly tasks requiring robotic assistance are elicited for initiating robotics engineer activity. Then, this robotic oriented mBOM is mapped within the proposed ontology. The latter actually requests all surrounding knowledge (i.e. working environment, available tools, initial/final part position, part mass, etc.) in order to infer robot trajectories as functional specifications, and payload and reachability as behavioural specifications with the support of an inference engine. Afterwards, these specifications are used in a computational tool embedding an A\* algorithm to generate both modular robot structure (structural design) and programme (logical design).

**Fig. 1.** Proposed knowledge-based methodology for modular robotic system design



### 3.1 Functional and Behavioural Design

In the preliminary steps of the proposed framework, engineering and manufacturing data from PDM and MPM systems are used and link to an ontology (i.e. knowledge base) that describes relations and rules for robot tasks definitions. The fact of combining ontology and an inference engine ensures the autocompletion of related knowledge of an actual context from an unconstrained problem. As such, Fig. 1 shows how a robotics engineer defines robot tasks such as ‘assemble with’ or ‘move from X to Y’. These descriptive needs are linked to CAD models, e.g. workpieces, working environment, that are also mapping to the ontology in order to infer important parameters for robot structural design. By using the pre-defined requests filled with input data from the robotics engineer and PDM and MPM systems, it is possible to infer, from the ontology, parameters such as the payload, movement range or even working rates and to compute the robot trajectory. This is doable because the ontology capture relations between assembly operations, robotic task, robot environment, robot action and motion planning.

For instance, if the robotics engineer defines a task corresponding to the following assembly operation ‘Assemble part-1 with part-2’, SPARQL requests will return the data related to the initial position, the assembly information of part-1 and part-2 (e.g. assembly sequence, orientation, etc.), and so on, until defining a set of subsequences of movements needed to achieve the desired task. These subsequences reflect trajectory discretization into points and orientation vectors with action such as closing a gripper or screwing. In addition, data such as mass, part tolerance, maximum distance to reach, are used for the structural configuration and programme computation as shown in Fig. 1.

### 3.2 Structural and Logical Design

The structural design of the modular robot is then based on a computational algorithm to elaborate a physical configuration, which fulfils the previously inferred functional and behavioural specifications. In such a context, an A\* search algorithm – based on a weighted graph which represents all possible module configurations – is proposed. Here each module could have zero or one rotational degree of freedom and a fixture system, allowing modules aggregation. By following the work done in [15], a similar computational strategy has been made, on which the weighted graph is built upon the functional specifications. Indeed, it is essential to adjust weight of the graph to orient A\* algorithm towards not only possible solutions, but also efficient solutions for the specific task. In other words, by penalizing some nodes  $n$  (i.e. representing a module) of the graph, it is possible to eliminate invalid solutions and put forward better solution. In such a context, nodes that minimize equation (1) will be preferred by A\* algorithm:

$$f(n) = g(n) + h(n) \quad (1)$$




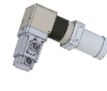

where  $h(n)$  is a heuristic that estimates the final cost of the design to achieve the desired goal (i.e. trajectory). The function  $h(n)$  returns the average error of the inverse-kinematic solver of the actual robot structure at node  $n$  for the discretized trajectory. The more error is higher, the more the node will be penalized. Collision detection analysis is also processed to check the geometric configuration. Finally,  $g(n)$  stands for the cost of a module, computed using a decision matrix (as represented in Table 1), where each criterion is pondered according to robotics engineer’s specifications. For instance, the equation (2) computes the allocated weight to a module at the tree stage  $S$  by dividing the estimated needed static torque for this node by the real static torque of the considered module:

$$A_i(S, M_u, D, \bar{m}, N) = W1 * \frac{M_u * D + (N - S) * \bar{m} * \frac{D}{2}}{T_m} \quad (2)$$

where  $M_u$  is the payload,  $S$  is the module’s stage in the tree,  $D$  is the distance from the tested module at the tree stage  $S$  and the desired goal,  $N$  is the maximum number of joint specified by the robotics engineer,  $\bar{m}$  is the average mass of available modules from the MPM system, and  $T_m$  is the static torque of the considered module. Therefore, if the real static torque is inferior to the estimated needed torque, the ratio will be

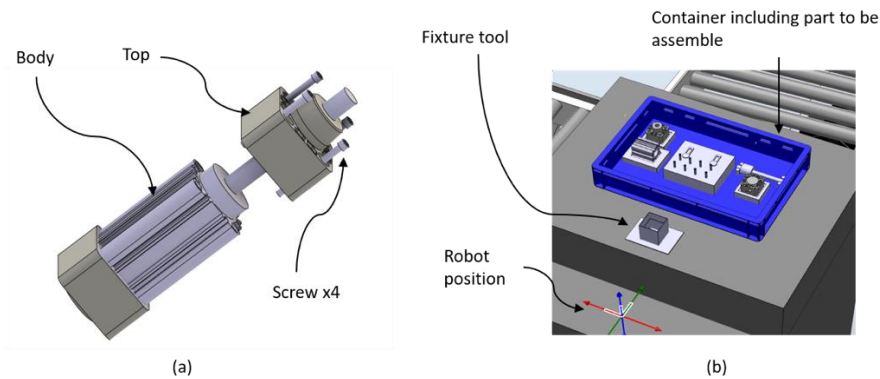
superior to 1, then penalizing this module. Finally, once the structural design achieved and checked, the logical design is addressed by synthesizing the trajectory and actions in simple robot instructions such as linear or joint motion with a time discretization.

**Table 1.** Decision matrix for the weighted graph computation

Module selection	Option					
	Module joint 1	Module joint 2	Module arm 1	Module joint 2	Module joint 3	
						
Criteria	Weight	Score				
Static Torque	W1	A1	B1	X	D1	E1
Accuracy	W2	A2	B2	X	D2	E2
....	...	...	...	...	...	...
Price	W4	A4	B4	C4	D4	E4
Speed	W6	A6	B6	X	D6	E6
Final Score		$\sum_{i=1}^n W_i A_i$	$\sum_{i=1}^n W_i B_i$	$\sum_{i=1}^n W_i C_i$	$\sum_{i=1}^n W_i D_i$	$\sum_{i=1}^n W_i E_i$

## 4 Case Study

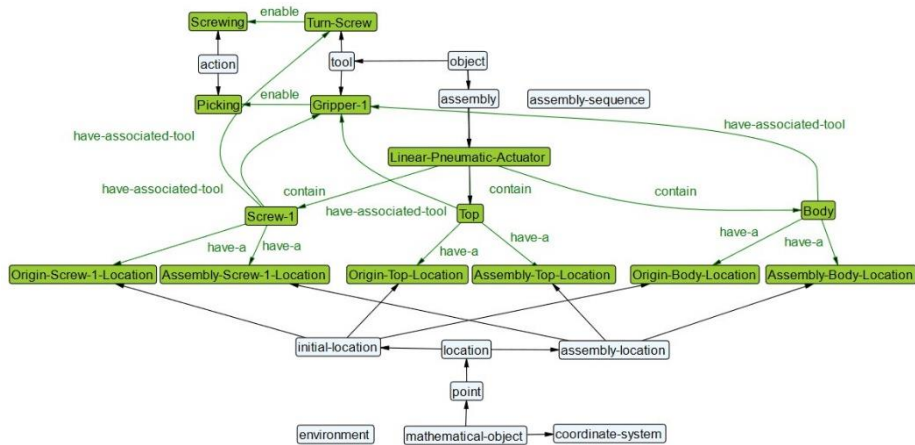
To illustrate the proposed methodology, a simple case study is introduced. The objective is to design a modular robot system to ensure a pneumatic linear actuator assembly. In this context, some assumptions have been made, for instance here the robot initial position is predefined, the end-effector orientation will remain constant for all operations.





**Fig. 2.** (a) Product assembly and (b) its working environment

The product to be assembled and the related working environment are described in Fig. 2. This information is stored in the PDM system (for product information) and the MPM system (for manufacturing information), then mapping within the developed ontology materialized by the concept graph in Fig. 3. According to the tasks' definition, pre-defined SPARQL queries (query language for searching into database) are executed to generate the relevant information such as assembly sequence, position and mass of each object to be assembled in the working environment. Finally, a list of minimal motions is established (i.e. composed of linear motions from point-to-point and actions such as 'close-gripper').



**Fig. 3.** Concept map and instances extracted from the ontology for robot task definition

For the sake of clarity, Figs. 4(a) and 4(b) give an example of SPARQL queries and returned results that are analysed to compute trajectories and parameters used for criteria evaluation using decision matrix describes beforehand. Finally, a weighted graph is pondered and the A\* algorithm is executed to generate both a design solution that fulfils functional and behavioural specifications, and a robot programme as illustrated by Fig. 5.

```
select ?Object ?Mass where
{?Object ?MassKg ?Mass.
Filter(?Mass > 0) }
```

Part	Mass (kg)
Screw1	0.05
Top	0.25
Gripper1	0.8
Body	0.75

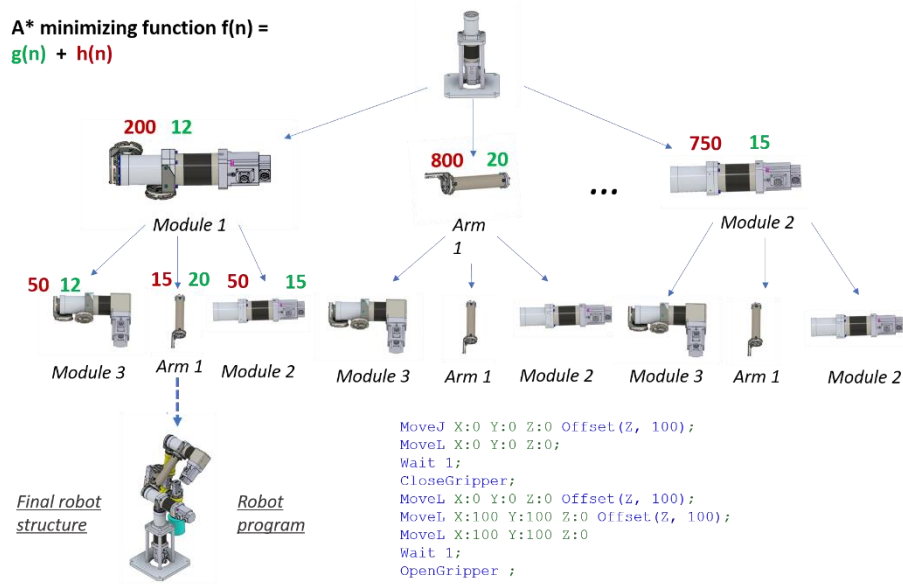
(a)

```
select ?Part ?Position where {?Part
?location ?Position . }
```

Part	Position
OriginBodyLocation	X:0 Y:0 Z:0
OriginTopLocation	X:10 Y:0 Z:0
OriginScrew1Location	X: 20 Y: 0 Z: 0
AssemblyBodyLocation	X: 100 Y:100: Z:100

(b)

**Fig. 4.** Example of SPARQL requests about (a) object's mass selection and (b) parts positions



**Fig. 5.** Weighted graph with preferred path highlighted the final modular robot structure and its related program

## 5 Conclusion and Perspective

A knowledge-based methodology for modular robot system design has been proposed in this paper. It is built upon two reasoning layers: (1) a knowledge-based processing layer for the identification of functional and behavioural specifications enabling robotics engineer to specify robotic tasks through abstract instructions, and (2) an A\* computational algorithm for generating both robot structure and program. Both layers have been implemented in a PLM ecosystem in which an ontology with an inference engine as well as a computational algorithm have been developed to generate robot specifications, structure and programmes. The case study has demonstrated the suitability of the proposed methodology for designing modular robots for task reconfiguration. However, currently, this model is not extensible to real world manufacturing operation. In fact, model information should be filled up to take into consideration important parameters such as assembly constraints, tolerance, robot calibration, etc. Moreover, the proposed methodology needs that all information about product assembly and manufacturing are stored in PLM and PDM systems, but these systems are not always implemented in SMEs and data are often incomplete and/or not up-to-date.

Built on these efforts, future work will be dedicated to the development of an application prototype using Unity 3D engine, ROS (Robot Operating System) and a laboratory prototype PLM system. Additional investigation will be focused on the

consideration of robot reconfiguration according to real-time data from manufacturing systems, therefore falling under the umbrella of context-aware cyber-physical systems design.

## Acknowledgements

The research activity has been made in collaboration with MS-INNOV company as part of a CIFRE contract (Industrial Standards of Research Training) by the French National Agency for Research Agency for Research and Technology (ANRT). The authors would like to thank all financial supports.

## 6 References

1. Gruninger, R., Kus, E., Huppi, R.: Market study on adaptive robots for flexible manufacturing systems. IEEE International Conference on Mechatronics, Malaga, Spain, pp. 1-7 (2009).
2. Spena, P.R., Holzner, P., Rauch, E., Vidoni, R., Matt, D.T.: Requirements for the Design of Flexible and Changeable Manufacturing and Assembly Systems: A SME-survey. *Procedia CIRP* **41**, 207-212. 2016
3. Matt, D.T., Modrák, V., Zsifkovits, H.: Industry 4.0 for SMEs: Challenges, Opportunities and Requirements. Springer International Publishing (2020.)
4. Multi Annual Roadmap for robotics in Europe, ICT24 Horizon 2020, (2015).
5. Hasbulah, M.H., Jafar, F.A., Nordin, M.H. : Comprehensive Review on Modular Self-Reconfigurable Robot Architecture. *International Research Journal of Engineering and Technology* **6**(4), 1317-1331 (2019).
6. Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular Self-Reconfigurable Robot Systems. *IEEE Robotics and Automation Magazine*, **14**(1), 43-52 (2007).
7. Spröwitz, A., Moeckel, R., Vespignani, M., Bonardi, S., Ijspeert, A.J.: Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robotics and Autonomous Systems*, **62**, 1016–1033 (2014).
8. Seeja, G., Arockia Selvakumar, A., Berlin Hency, V.: A Survey on Swarm Robotic Modeling, Analysis and Hardware Architecture, *Procedia Computer Science*, 133, pp. 478-485, ISSN 1877-0509 (2018).
9. Aloupis, G., Collette, S., Damian, M., Demaine, E.D., Flatland, R., Langerman, S., O'Rourke, J., Ramaswami, S., Sacristán, V., Wuhler, S.: Linear reconfiguration of cube-style modular robots. *Computational Geometry*, **42**, 652–663 (2009).
10. IEEE Standard Ontologies for Robotics and Automation," in *IEEE Std 1872-2015*, pp.1-60, (2015).
11. Bi, Z.M., Lin, Y., Zhang, W.J.: The general architecture of adaptive robotic systems for manufacturing applications. *Robotics and Computer-Integrated Manufacturing*, **26**, 461–470 (2010).
12. Fitch, R., Butler, Z., Rus, D.: Reconfiguration planning for heterogeneous self-reconfiguring robots, *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, USA, pp. 2460–2467 (2003).

13. Desai, R., Safonova, M., Muelling, K., Coros, S.: Automatic Design of Task-specific Robotic Arms. arXiv preprint:1806.07419 (2018).
14. Valente, A.: Reconfigurable industrial robots: A stochastic programming approach for designing and assembling robotic arms. *Robotics and Computer-Integrated Manufacturing*, **41**, 115–126 (2016).
15. Chandrasegaran, S.K., Ramani, K., Sriram, R.D., Horvath, I., Bernard, A., Harik, R. and Gao, W.: The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design*, **45**(2), 204-228 (2013).
16. Ramos, F., Vázquez, A.S., Fernández, R., Olivares-Alarcos, A.: Ontology based design, control and programming of modular robots. *ICA 25*, 173–192 (2018).
17. Tenorth, M., Beetz, M.: Representations for robot knowledge in the KnowRob framework. *Artificial Intelligence*, **247**, 151–169 (2017).
18. Demoly, F., Kim, K., Horváth, I.: Ontological engineering for supporting semantic reasoning in design: deriving models based on ontologies for supporting engineering design. *Journal of engineering design*, Vol. 30, 405-416 (2020).
19. Demoly, F., Yan, X.-T., Eynard, B., Rivest, L. Gomes, S.: An assembly oriented design framework for product structure engineering and assembly sequence planning. *Robotics and Computer-Integrated Manufacturing*, **27**(1), 33-46 (2011).
20. Demoly, F., Troussier, N., Eynard, B., Falgarone, H., Fricero, B., Gomes, S.: Proactive assembly oriented design approach based on the deployment of functional requirements. *Journal of Computing and Science in Engineering* (2011).