Capability-based Frameworks for Industrial Robot Skills: a Survey

Matteo Pantano, Thomas Eiband, Dongheui Lee

Abstract—The research community is puzzled with words like skill, action, atomic unit and others when describing robots’ capabilities. However, for giving the possibility to integrate capabilities in industrial scenarios, a standardization of these descriptions is necessary. This work uses a structured review approach to identify commonalities and differences in the research community of robots’ skill frameworks. Through this method, 210 papers were analyzed and three main results were obtained. First, the vast majority of authors agree on a taxonomy based on task, skill and primitive. Second, the most investigated robots’ capabilities are pick and place. Third, industrial oriented applications focus more on simple robots’ capabilities with fixed parameters while ensuring safety aspects. Therefore, this work emphasizes that a taxonomy based on task, skill and primitives should be used by future works to align with existing literature. Moreover, further research is needed in the industrial domain for parametric robots’ capabilities while ensuring safety.

Index Terms—PPR, HMLV, task, skill, primitive, robot, review, survey

I. INTRODUCTION

Detailed a-priori planning of manufacturing processes defined in nowadays industry is going to be soon outdated with “high mix - low volume” (HMLV) manufacturing driven by heterogeneous demand for product variants [1], [2]. Therefore, capability-based engineering envisioned in Industrie 4.0 is slowly entering the domain of manufacturing to ensure business continuity [3]. Through this concept, factories of the future (FoF) will be able to adapt their production plans during order execution as long required capabilities will be used to describe production processes instead of actual resources [3]. However, for the implementation of capability-based production, resources (e.g., robots, CNC machines) will need to provide descriptions of their capabilities (e.g., 3-axis milling, Cold Metal Transfer (CMT) welding) to ensure correct planning by linking their capabilities with manufacturing requirements [3]. One approach to link resources to requirements was initially introduced in the standardized Product Process and Resources (PPR) model described in [4].

Since then, different Product Lifecycle Management (PLM) systems started using the model for their simulations. However, such definitions have not widely reached the robotics community and different varieties of definitions and nomenclature have been proposed [5]. Therefore, an alignment between robotic research literature and PPR literature is needed to overcome one of the adoption barriers of capabilities in manufacturing [6]. To bridge this gap, this research presents a structured literature review which focuses on definitions used when describing robots’ capabilities considering industrial scenarios. More specifically, this work aims to answer the following research questions:

RQ1: Which nomenclature is most frequent in robotics when describing capabilities? And with which taxonomy?
RQ2: What are the most investigated robots’ capabilities?
RQ3: What distinguishes industrial robotics applications using robotic capabilities from academic ones?

To provide a clear description of the research done for answering these questions, this work is structured as follows. At first, in Sec. II, definitions for a capability based architecture are given. Second, in Sec. III, the structured review criteria are outlined. Third, in Sec. IV, the results from the review are presented. Finally, in Sec. V and VI, the conclusions with future outlooks are given. Moreover, all the data used for this review is available in the supplementary on GitHub1.

II. ARCHITECTURE AND DEFINITIONS OF TERMS

To conduct the research, the systematic literature review approach defined by [7] was employed. Therefore, categories for the classification had to be defined. In this section, such terms are defined. The designated expressions come from two research topics. On one hand, from the manufacturing domain where the concept of Plug-and-Work based on PPR [8] describes capabilities at the shop floor. On the other hand, from the robotics domain, where ontologies have been defined to represent robots’ capabilities necessary to solve complex steps like the assembly of a chainsaw as described in the FoF ontology [9]. This resulted in the architecture shown in Fig. 1.

A. Process

A Process is defined as an ensemble of different Skills and depicts an abstract description of steps in a workflow to reach a certain desired outcome as defined by [5], [11]. Definition of a Process finds its roots in the definitions of the enterprise-control system integration in the well known ANSI/ISA95 [13]. However, in this context the definitions of Process as defined in PPR is used. Therefore,

1https://github.com/eiband/industrial-skill-review
a Process is solution neutral and its execution depends on the type of resources involved and their capabilities [5], [8].

B. Task

A Task is defined as an ordered ensemble of different Skills and depicts a concrete representation of steps in a workflow to solve a specific goal by interfacing with operators, control systems and programs [14]. Therefore, it could be seen as a more specialized version of a Process. For the sake of clarity, a Task can be easily described as a sequence of Skills that have been properly parameterized upon the resources involved and their capabilities.

C. Skill Group

A SkillGroup is defined as a collection of Skills, which allows to group similar ones together. Such grouping has been used in [5], [11] to structure a large variety of Skills into meaningful groups that are understandable to the user (e.g. move, connect, compare). The SkillGroup is not considered during execution, but it has a descriptive character when a user is searching for available Skills.

D. Skill

A Skill is a predefined robot’s capability that can be parameterized to solve a specific goal. A Skill can be either a physical capability or a perception capability [14]. Skills that execute physical actions are able to alter the physical world state, for example picking an object. Skills with perception capabilities can update only a world representation based on the made observations but do not alter the physical world. An example is the measurement of an object’s pose.

E. Parameterized Skill

A ParameterizedSkill can be the instance or be implemented as inherited class of a Skill equipped with parameters that are Task and resource specific, hence, it can be executed on robot hardware to accomplish a goal.

F. Parameter

Parameters are used to configure a Skill for a specific Task [10], [15]. Parameters can be specified by different methodologies, for instance manually defined and interpretable by the user, or automatically extracted by the system and non-human interpretable [16]. This difference is denoted by calling them respectively Parameter and DerivedParameter.

G. Primitives

A Primitive is the closest atomic unit to the hardware-level, also know as atomic function that can perform a distinct operation. It can be depicted as building block when composing Skills, for instance opening a gripper [14], [17]. Similarly to Skills, Primitives can be parameterized to solve a precise task and could provide output information, for example, the location of an object. Additionally, Primitives can be grouped as Skills in SkillGroups, however, this is not considered in this survey.

H. Example of how the architecture can be used

To demonstrate the proposed architecture an example of an automation process that is solved by a robotic task and how it could be depicted using the above nomenclature is here presented. The example is shown in Fig. 2. Imagine that the Process of a gearbox assembly shall be automated. Therefore, the user identifies, via a programming method (e.g., learning from demonstration), appropriate robot Skills (i.e., Pick and Place, Pick and Insert and Pick and Screw) from different SkillGroups, such as Manipulation. Afterwards, to execute it, resources are matched to the Process via a task planning algorithm or by a capability-based manual assignment on a Manufacturing Execution System (MES) considering that a Transmission gearbox needs to be assembled. Therefore, the actual gearbox assembly becomes a Task consisting of a sequence of ParameterizedSkills. These ParameterizedSkills are the instances of the Skills equipped with their parameter values. For example, Pick and Place(Shaft, Housing) denotes a Pick and Place skill that involves the SHAFT and HOUSING Parameters which are digital artifacts representing properties of the physical objects. The information within these artifacts is then used to assign parameters to the underlying Primitives, for example Move(target=pos_4). Furthermore, a Move primitive could also have a DerivedParameter as used within the Pick and Insert skill, for example the Dynamic Movement Primitive (DMP) [18] weights of the represented motion, Move(target=pos_4, traj=dmp_weights).

III. REVIEW PROCESS

Due to the research bulk on the topic, this review used a systematic research method (SRM) as outlined in Sec. II. This section highlights how the expression previously defined were used along with exclusion criteria, search strategy and the research protocol.

A. Literature search

To collect candidate papers, an automatic search on the Scopus digital library database was performed on 6th October 2021. The search terms targeted the industrial usage

2https://www.scopus.com/
of robots’ capabilities published between 2014 and today as long this time span presented the largest amount of research publication on the field. This resulted in the following Scopus search string:

\[
[(\text{robot AND skill}) \text{ OR (robot W/15 skill)}] \\
\text{AND (industrial OR manufacturing)}
\]

The query was applied to the research fields: title, abstract and keywords and resulted in a total of 210 papers.

B. Selection

Afterwards, an exclusion criterion on the abstract and title, filtering out papers that were not fitting due to topic irrelevance was applied (i.e., skills in the workforce required for usage of robotics) and 149 papers were discarded. The remaining 61 papers were fully read and analyzed. During the process, 27 other publications were added as long they were describing relevant previous works of authors identified in the previous step. This resulted in a total of 88 fully analyzed papers.

C. Classification

To understand how and which nomenclature the research papers used to define robotic capabilities, these classification criteria were created:

- **Skill model.** Evaluation whether the authors define what a skills is and how a skill model is structured.
- **Similarity.** Understanding if the proposed capability-based skill framework is similar to the one presented in order to evaluate the proposal of this work. This criteria recorded if the framework showed the same structure as the one presented in this work.
- **Industrial.** To know if the research was more industrially focused or is not important to understand the technical readiness level (TRL) of the technology. Therefore, it was marked if the research work was conducted on a use case of a real-manufacturing scenario.
- **Industrial requirements.** Knowing if the requirements for industrial application that are necessary to enter a specific market are met is another important insight to understand the TRL of the technology. Considering that large amount of the literature on capabilities is from Europe [6], the criteria for accessing the European market were used to assess the development level of the frameworks \(^3\) (the full list of the requirements can be found in the supplementary material).
- **Implementation.** Knowing the implementation technologies is important to understand the applicability in other scenarios. In this term, the frameworks and programming languages were recorded if implementation details were provided.
- **Parameters.** Assigning parameters to skills enables generalization capabilities. If parameters were used, their type was reported.
- **Definitions.** To understand how the definitions provided in Sec. II were used, the nomenclature used in the reviewed works was mapped to the definitions provided above using a review table (the full review table can be found in the supplementary material).

IV. ANALYSIS OF THE RESULTS

To analyze the results the semantic properties and frequencies of the terms were analyzed. This section describes the results obtained from this analysis.

A. Classifications results

By applying the classification criteria on the 88 papers the following results were obtained:

- **Skill model.** 26 papers out of 88 proposed a clear skill model used in their skill framework.
- **Similarity.** 57 papers out of 88 used a capability-based skill framework similar to the one proposed in Sec. II.
- **Industrial.** 45 papers out of 88 were dealing with an industrial use case.
- **Industrial requirements.** 61 papers out of 88 considered some of the requirements needed for industrial usage.

\(^3\)Similar requirements however, apply also to other markets
• **Implementation.** 49 papers out of 88 clearly explained the used tools and frameworks for the implementation.

• **Parameters.** 32 papers out of 88 defined and explained the parameters used for their skill frameworks.

• **Definitions.** Considering the definitions in Sec. II, the research papers could be summarized as follows. The categories which had the most amount of information were skill (79 out of 88), task (65 out of 88) and primitives (49 out of 88). The remaining categories were used much less frequently, skill group (14 out of 88), parametrized skill (17 out of 88) and process (31 out of 88).

### B. Nomenclature

Within the definitions in Sec. II, also the types of skills, tasks and primitives were recorded. To study which names were most common across the research works, and provide data for RQ1 and RQ2, the data was preprocessed and the most common terms identified.

1) **Preprocessing:** In order to prepare the extracted data for clustering, a number of preprocessing steps to the manually extracted definitions obtained from the analysis performed in Sec. III were applied. For the sake of clarity the denomination of a task, skill, or primitive is defined as label in the following paragraphs. First, the labels from the review table under the definitions column were extracted. Whenever authors provided labels in camel case, they were resolved to words with underscores, for instance *MoveTo* resulted in *Move_To*. Next, labels were converted to lowercase. Then, lemmatization was applied on each of the words. Here, inflectional endings were removed, i.e., "moving" would result in "move". Hereby, the WordNetLemmatizer from the natural language toolkit (NLTK) [19] was used and 526 labels were obtained for the subsequent steps.

2) **Identified Common Terms:** A search about common terms was applied using a wordcloud[^4] based on the label’s frequency (bar plots are also available in the supplementary materials). The results for task, skill and primitive are visualized in Fig. 3. The naming task, skill and primitive are the most used by the research community therefore answering to the first part of RQ1. However, other nomenclatures like action seem to be frequently used in robotics [20]. Moreover, the most investigated types were: *assembly* for task (also in line with the identified most required capability by [21]), *pick* and *place* for skill and *motion_primitive*, and *open_gripper* for primitive.

3) **Semantic similarities with K-means clustering:** To investigate if researchers had a similar focus on action types, a K-means clustering was applied after removing duplicate terms. This section reports the cumulative analysis on primitives, skills and tasks. Initially, the terms were encoded in feature vectors using sentence transformers (SBERT) [22] with the pre-trained model all-microsoft-base-v2, known for its good performances in general purpose tasks. Afterwards, a K-means clustering on the encoded features was applied with the parameters of 10 clusters and dimensionality reduction to 2 for visualization purposes. Finally, for each of the obtained clusters, a keyword search was applied. The two words scoring the best cosine similarity with all the words present in that cluster were identified. The identified clusters are denoted in Table I (visual results are also available in the supplementary material). From these results, the following insights can be drawn. Firstly, it can be perceived that the research focuses mostly on the group pick-placement, therefore answering to RQ2. This can be related to the tasks that industrial use cases commonly face [23], [24]. The groups motion-movement and grip-gripper are implemented by the researchers mostly as primitives like in [25], underlining that those simple capabilities are the building blocks to create different skill types. This is also reflected by the most occurring primitives (motionPrimitive, open_gripper) identified in Sec. IV-B2. The groups button-press, clean-wipe, navigate-circular, object-registration, placement-pick, rotate-spin and spray-paint are mainly implemented by the researchers in the skill level like in [26] and they represent several robots’ capabilities necessary to accomplish tasks (i.e., clean-wipe for the task of cleaning a room). Finally, the machine-code group is integrated in the task level due to the large amount of necessary capabilities when robots have to interface with machines. For example in [10], the task machine feeding is identified where the robot should be capable of interacting with the machine (e.g., set inputs/outputs) and handle objects of different sizes and shapes (e.g., picking, placement, locate).

### C. Industrial and non industrial scope

A frequency analysis was performed to identify most common terms in the two sub-sets given by filtering the *Industrial* criteria for identifying data regarding RQ3. The overall analysis can be seen in Fig. 4.

1) **Implementations:** Industrial scenarios show a diverse set of frameworks closer to the automation domain (i.e. programmable logic controller (PLC) language, AutomationML (AML)) [5], [32]. Additionally, the Robot Operating System (ROS) also finds its way into such scenarios [14], [87]. In comparison, non-industrial applications rely quite often on ROS with the python programming language, such as in [25]. Hereby, ROS has the main purpose to serve as a communication middleware between so-called nodes but it also defines interfaces in the form of standardized message formats. However, ROS is not used to implement knowledge itself, which is an important requirement for non-industrial applications. Thus, some works rely on ontological representations. Ontologies can be implemented in the W3C Web Ontology Language (OWL), and some of them were already standardized, such as the IEEE 1872 Core Ontology for Robotics and Automation (CORA) first proposed in [88] and validated in [89].

2) **Industrial requirements:** The requirements regarding type of hardware used, software version and robot intended behaviour were equally considered both for the industrial and the non-industrial scenarios like in [63], [86]. The major difference is that some of the industrial scenarios consider also the requirements related to safety aspects of the application like in [38], [73]. This has been always a major point of difference between industrial and non-industrial research [21], and this is reflected also in this review concerning skill frameworks. Therefore, to increase market adoption of skill frameworks in the industrial domain, safety should be addressed either by having inherently safe skills or by conducting a risk analysis behind each robotic skill.

[^4]: http://amueller.github.io/word_cloud/
speed. These parameters
industrial oriented
Primitive
- (A1), skill (B1) and primitive (C1). On the bottom the most referred tasks (A2), skills (B2) and primitives (C2). The most common task was assembly, the most common skills pick and place and the most common primitives motion_primitive and open_gripper.

Table I: Review table associated to the different works. The table shows the classified research works on the cluster level and on the robot’s capability complexity. From the table it is easy to perceive that the placement-pick group is the most investigated by the researchers.

| Cluster name       | Task                  | Skill                  | Primitive             |
|--------------------|-----------------------|------------------------|-----------------------|
| button-press       | [27], [30], [39], [40] | [15], [15], [15], [15]| [20], [40], [20], [20] |
| clean-wipe         | [50], [54]            | [24], [24], [24], [24]| [20], [20], [20], [20] |
| grip-gripper       | [41], [39], [37]     | [12], [12], [12], [12]| [12], [12], [12], [12] |
| machine-code       | [40], [39], [39], [39]| [50], [30], [30], [30]| [50], [50], [50], [50] |
| motion-movement    | [70], [40], [40], [40]| [50], [50], [50], [50]| [50], [50], [50], [50] |
| navigate-circular  | [30], [30], [30], [30]| [50], [50], [50], [50]| [50], [50], [50], [50] |
| object-registration| [15], [15], [15], [15]| [15], [15], [15], [15]| [15], [15], [15], [15] |
| placement-pick     | [41], [39], [39], [39]| [50], [50], [50], [50]| [50], [50], [50], [50] |
| rotate-spin        | [20], [14]            | [15], [15], [15], [15]| [15], [15], [15], [15] |
| spray-paint        | [30], [30], [30], [30]| [50], [50], [50], [50]| [50], [50], [50], [50] |

3) Parameters: In the industrially focused works, the parameter scope of a skill is closer to hardware functions and physically measurable data. The most used parameters were height, position, offset, and robot_speed. These parameters appear to allow only minor adaptions to the robotic task and appear to take rather hardcoded values such as the absolute position of an object provided by the programmer [14], [87]. In comparison, the complexity of parameters in the non-industrial scenarios is considered to be higher. The most commonly used parameters were goal, target, world, and robot. Parameters of this scope allow major modifications of the skill’s behavior. A skill that is parameterized with an abstract target instead of an absolute position could adapt its behavior significantly by exploiting further information from a knowledge base like in [11], [34]. Parameters can be also seen as function arguments that can be either passed to a skill or a primitive. Considering different programming layers, parameters would describe the input ports of a skill visible to the user, while parameters could also describe the function arguments of a primitive, which are visible only

Figure 3: Wordcloud representing the occurrence of words in the classification table. On the top the names given for task (A1), skill (B1) and primitive (C1). On the bottom the most frequently appearing words.
to the system designer. In both industrial and non-industrial cases, the parameters were mostly found to be associated with a skill. The internal logic of the skill is then meant to extract meaningful values that are shared with the underlying primitives. Examples of this structure can be also found in [15], [43].

With these findings it is possible to find the answers to RQ3. Industrial usage of robot capabilities distinguishes itself from non-industrial usages on two areas. First, its focus is on simple skills with often hard-coded parameters. Second, the implementation uses frameworks close to the automation domain while always respecting the safety of the application.

D. Approaches to capability-based skill frameworks

Finally, from the review it was possible to see that 57 out of 88 papers used a similar architecture as the one proposed here. Therefore, a tree-like structure where primitives are the closest units to the hardware level and the tasks the farther away from the hardware seems to be a concept that most researchers agree on, both for the industrial and the non-industrial cases. Therefore, answering to the second part of RQ1. The best examples on the usage of the identified architecture can be found in [14], [15], [65]. In these works, a common skill model is presented. Often this model is dependent on the resource which provides certain functionalities (i.e., primitive) and the input/output variables which can parameterize the functionality. The best example of such modelling can be seen in [5].

V. TRENDS AND OUTLOOKS

During the review, the necessity to accommodate market demands leading to HMLV productions has been underlined as also identified by [6]. To adapt automation in such production scenarios, robots with skill frameworks were seen as enabling technology within Industrie 4.0 [90], [91]. The aim of this technology is to avoid the high costs of manual processes on the one hand and the limitations of fully automatic, poorly customizable processes on the other hand. To properly exploit the advantages of skill frameworks, however, skill hardware and vendor independence is a key factor as long as it guarantees wide skill applicability [8] and, for example, skills could be used across multiple plant sites of manufacturing companies [49]. To enable such independence, primitives will need to be properly mapped to skills according to the available hardware functionalities. Therefore, an automatic primitive to skill mapping is worth investigating [61]. However, such mapping would require an universal information representation among all employed skills [51]. To lay a foundation for that, skills and their primitives could be defined in industrial standards such as AML like in [5] or definitions like the FoF ontology [9]. Within this aspect, it is worth noting that industrial applications initially preferred AML as information representation and this is also visible in Fig. 4. However, in the last years, OPC UA has become more common [15] and none of the surveyed works report to use AML in the last three years. Apart from skill definitions, parameters are important to enable skills’ reusability. In many industrial applications, skill parameters are still manually defined [38]. However, recent works consider automatic parameterization techniques, where the skill sequence and skill parameters are defined either by an autonomous planner or extracted from human demonstrations [27], [28]. Also, the complexity of parameters is changing. From simple, physical quantities such as positions, parameters are moving towards more abstract ones, such as object IDs or even interfaces to world models which are passed as parameters [5]. This shows that the responsibility of interpreting a parameter is being shifted from the human to the skill itself.

VI. CONCLUSIONS

This work presented the review of several papers on the field of capability-based skill frameworks by focusing on the robotic industrial domain. The review was performed via a structured approach and the research works were classified according to some predefined criteria. The results obtained were then analyzed using semantic clustering or frequency of appearance. Through this methodology, the following results were obtained. Firstly, the analysis showed that the research practitioners, when referring to capabilities, use often the names task, skill and primitive, where primitive is the closest to hardware, tasks the furthest and skills represent robots’ capabilities. Secondly, several research areas defined by the type of robotic capability have been identified. From this classification it was discovered that pick-placement is the most researched capability and that motion-movement, gripper are common groups of primitives used to create different skills. Finally, some differences have been found between industrial (I) and non-industrial (NI) research. I uses parameters that are close to the hardware, whereas NI uses high level ones. Considering implementation frameworks, I prefers PLC languages while NI others (e.g., ROS). For industrial requirements, it was found that I considers more the safety aspects when compared to NI. Apart from these main findings, the review also showed an increasing interest on the usage of robots and skills to accommodate requirements of a HMLV production and that information representation is essential to enable skill reusability, either via OPC UA or other standards. While performing the review, a major pitfall was identified on the research query. Such query was largely biased towards industrial scenarios and might better represent this domain compared to non-industrial one. Therefore, future work could focus more on the review of purely academic works in robots’ capabilities to give a bigger picture of the field.

REFERENCES

[1] G. Anshuk, C. Magar, and R. Roger, “How technology can drive the next wave of mass customization: Seven technologies are making it easier to tailor products and services to the wants of individual customers—and still make a profit.” 2014.
[2] M. Fechter, C. Seeber, and S. Chen, “Integrated process planning and resource allocation for collaborative robot workplace design,” Procedia CIRP, vol. 72, pp. 39–44, 2018.
[3] A. Bayha, J. Bock, B. Boss, C. Diedrich, and S. Malakuti, “Describing capabilities of industrie 4.0 components,” [online] Available: https://www.platform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Capabilities_Industrie40_Components.html (accessed on 2022/04/24), p. 33, 2020.
[4] A. F. Cutting-Decelle, R. Young, J. J. Michel, R. Grangel, J. Le Cardi- nal, and J. P. Bourey, “Iso 15531 mandate: A product-process-resource based approach for managing modularity in production management,” Concurrent Engineering, vol. 15, no. 2, pp. 217–235, 2007.
[5] J. Backhaus and G. Reinhardt, “Digital description of products, processes and resources for task-oriented programming of assembly systems,” Journal of Intelligent Manufacturing, vol. 28, no. 8, pp. 1787–1800, 2017.
