A Formal Skill Model Facilitating the Design and Operation of Flexible Assembly Workstations

Lauren van De Ginste, Alexander De Cock, Axl Van Alboom, Yogang Singh, El-Houssaine Aghezzaf, Johannes Cottyn

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.

A formal skill model facilitating the design and operation of flexible assembly workstations

Lauren Van De Ginste[1,2][0000−0001−9087−6329], Alexander De Cock[3], Axl Van Alboom[1,2], Yogang Singh[1,2], El-Houssaine Aghezzaf[1,2], and Johannes Cottyn[1,2]

[1] Department of Industrial Systems Engineering and Product Design, Ghent University, Technologiepark 46, 9052 Gent-Zwijnaarde, Belgium, www.UGent.be
[2] Industrial Systems Engineering (ISyE), Flanders Make vzw, www.FlandersMake.be
[3] CodesignS, Flanders Make vzw, Belgium, www.FlandersMake.be

Abstract. In the Industry 4.0 area, there is an increasing demand for highly customized products in small batch sizes. Final assembly operations are frequently targeted to embed flexibility and compensate for the growing manufacturing uncertainties. Therefore, an adequately designed and operated flexible assembly workstation is crucial. Converting the flexibility needs into design and operational decisions requires versatile formal models delivering generic descriptions of needs and capacities. Skills form the central connector between products, processes and resources. Here, a skill-centered model for describing resource activities, the related production needs and flexibility impacts is introduced. The model fits both plug and produce and design optimization settings and goes beyond current skill-based modelling by offering a framework which, by design, does not limit the applications and easily adapts to the desired level of detail. One key strength is its ability to combine abstract and executable skills. Next to the product-action skills, also assistive skills related to operator support, parts storing, ergonomics etc. can be easily modelled. The use of the model is illustrated by an example based on an industrial use case from Flemish industry.

Keywords: Assembly workstation flexibility · Skill-based engineering · Resource modelling

1 Introduction

In today's highly competitive environments, manufacturing deals with a lot of challenges. One of them is increasing flexibility while keeping cost low and quality high. Final assembly operations are frequently targeted to embed flexibility and compensate for the increased manufacturing uncertainties and customization [15]. The individual resources and configurations in an assembly workstation setting directly impact the overall plant-level flexibility through base flexibility dimensions relating to the key components of a manufacturing company (personnel, equipment and material) [17]. Also, not only directly value-adding resources have a high impact, but also resources related to operator support, part storing, ergonomics etc. have an not to be neglected effect.

In [9] it is recognised that a common formal capability-based resource model is needed to allow rapid system design and reconfiguration decisions in heterogeneous multi-vendor production environments. This model should allow a well thought-out design, objective resource selection and further reasoning during operations and reconfigurations [5]. A lot of research in manufacturing is focused on the development of models to describe products with information as design, geometry and required processes. A standardized and universal applicable model for resources producing or assembling these products is frequently neglected and application- or vendor-specific [18]. In [7] the potential and importance of formal ontologies to capture and share assembly knowledge is illustrated. They support collaboration across the assembly design and process planning domains. Because skills are identified as a connector between products, processes and resources, the here-introduced model could form a building-block between product requirements, assembly sequences, resources, operations and their impact on flexibility in an assembly knowledge store [1].

The paper is organised as follows. Section 2 describes the current state-of-art in skill-based assembly resource modelling. Section 3 introduces the formal skill model and illustrates it's use and key strengths with some clarifying examples. Finally, conclusions and further research directions are found in section 4.

2   State-of-art

Several research projects have recently looked into skill- or capability-based modelling in the manufacturing and assembly domain. Before further elaborating on the current state-of-the-art (section 2.2), the different viewpoints on the meaning and use of the term 'skill' versus 'capability' are highlighted (section 2.1).

2.1   Capabilities versus skills

It is recognised in literature that, for manufacturing and assembly, resources can be best described by their skills or capabilities which mimic their capacities and ability to complete tasks [11]. Resources in manufacturing are seen quite broad and can be any physical entity, representing hardware and software, that is involved by the execution of a process [5]. This includes equipment, operators, parts, sub-assemblies etc.

The term 'capability' was first used in literature together with the term 'competence' to describe high-level human qualifications in manufacturing. However today it is also frequently used in more general formal models for the manufacturing domain [11]. A capability is in this context defined by [6] as "the inherent ability of the manufacturing system to perform change in the properties of a material in order to produce tangible products". According to [8] this definition falls short on not including assembly operations and those capabilities which don't change the properties of the product but are required during production. Capabilities however remain closely related to adding value to products [6].

The term 'skill' is sometimes used interchangeably with 'capability' but is mostly more focused on executable machine functionalities working bottom up [8,11]. In [14] skills are identified in the robot programming context as "re-occuring actions that are needed to execute standard operation procedures". In [10] the term skill is specifically used for the superiority features of an operator defining it's worker performance in performing tasks. More broadly, skills are seen as a connector between products, processes and resources enriched with additional information [1,14]. A skill is also very broadly defined as "the potential of a manufacturing resource to achieve an effect within a domain [12]". A lot of overlap is found between both terms. Small connotation differences can be identified but no real consensus can be found in literature. Skills seem to be more frequently used on the lower execution level than capabilities, while capabilities rather start from value-adding activities.

The focus of the here-introduced model is broad and in fact spans both terms. Nevertheless, one term should be chosen. Because a more bottom-up approach is envisioned allowing higher levels of detail for execution activities (but at the same time also more abstract design activities), the term 'skill' will be used.

2.2 Skill-based modelling

When looking into the development of skill-based models, Kocher et al. [11] states that it is important to keep the model as general and open as possible to enable reuse and future extensions. Next to models solely focusing onto the skill- or competence-level of operators, two main categories of skill-based modelling approaches can be identified [11,10].

The first category are formal models such as ontologies to create skill structures that can be used as a shared vocabulary or for optimization and reasoning purposes. Most contributions start from the process or task that is translated into a required skill [9,3,14,4]. Järvenpää et al. [9] developed a model to support 'capability matchmaking' between product requirements and resource skills. Later on it is extended with 'executable capabilities' for task programming but without a direct link between both [16]. Hashemi-Petroodi et al. developed a three-step methodology to automatically reconfigure reconfigurable manufacturing systems (RMS) [4]. Feasible assembly plans are set-up using a skill-based comparison of the product requirements and RMS. The skill comparison is based on a semantic matching process. The assigned resources are only afterwards checked through a quantitative parameter analysis. Hoang et al. [5] proposed as one of the first a more product-oriented approach to limit modelling and communication effort during reasoning. In their model the skills of resources are represented as ranges of product property values. However, the efforts needed to deduce the important properties and keeping them updated is a disadvantage compared to the process-oriented approach. In [11] a formal model of machine skills that directly includes both a description of high-level and executable skills is presented. Their model is an alignment ontology, built upon different sub-ontologies based on industry standards. Because of the interlinking between various sub-ontologies, modelling

is complex and does not adapt quickly to the desired level of detail. In [13] equipment is uniformly identified via high level skills which can be matched with the product-specific manufacturing requirements embedded in the product's bill of processes. From these high-level descriptions, executable skills are derived and made available as services.

The second category focuses on the execution level including plug and produce environments where skills are used to encapsulate machine functionalities for task-level programming [11]. In [1] a similar concept as in [13] is implemented, but more focused on task-oriented programming. Here, skill descriptions are used for the vendor-independent modelling of resources, processes and products. Their taxonomy is mainly built on 5 top-level skills which can be subdivided into more specific skills. Danny et al. [3] actively developed an openly accessible plug-and-produce system using AutomationML and OPC UA. The concept of skill recipes and skill requirements is used to execute the resource skills to fulfil the assembly requirements. Rovida et al. [14] focus on an even more narrow scope and specifically targets task level robot programming by providing robots with a set of movement primitives on the controller level and skills operating as standard operating procedures. The skills are implemented by the primitives which are mostly inferred through autonomous reasoning by the robot.

All aforementioned approaches differ in their level of granularity, implementations, applications and modelling languages which makes it difficult to merge and reuse them. The only common ground between all above models is the uncoupling between products and resources via skills. However, none of them is application independent and easily adapts to the desired level of detail.

3   Formal skill-model

The proposed skill-centered model describes and links resource capacities to the related production needs via skills showing their impact on the 'world'. The model is meant to be versatile and can be used during both the design and execution phase to suggest reconfigurations or other optimisations. It is perfectly suited for skill-based programming and scheduling of machine functionalities, but also for the matchmaking between tasks and resources including constraints. Futhermore it can link to operational data and serve as a base to analyse operations and asses the performance and robustness. It is meant as a core model which connects with different other models (resource model, skill condition model and task model) and allows for application specific extensions.

3.1   Conceptual model

Figure 1 shows the conceptual skill-model. It is displayed using the Unified Modeling Language (UML) notation to deliver an abstract view on the high-level meta-model. This way, it easily shows the links between different aspects and aids for easy problem understanding, analysis and documentation [2]. Different software implementations are possible. Because of the high-level of interlinking,
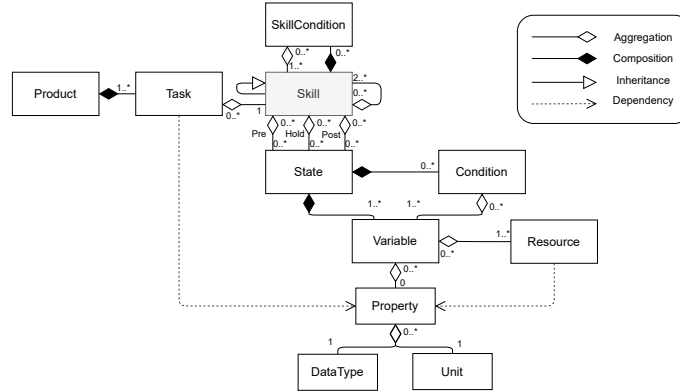
Fig. 1. Conceptual skill-model, represented in UML

ontology based graph databases seem the best suited for large examples and data storage (Section 3.2).

Skills are the central element describing the applicable transitions between states. The model recognizes the four dimensional nature of the skill concept. Skills can be atomic or composed, process-independent or process-specific, product-independent or product-specific and even resource-independent or resource-specific [12]. First, it is easy to understand that skills can be decomposed into subskills until they cannot be decomposed any further. When this atomic point is reached, depends on the required level of granularity and is thus application specific. An example is the skill 'transport' that is composed of a 'pick' and 'place' skill. However, composition is also closely related to refinement, if different compositions are possible, a refinement is required. Refinements add detail to the skills based in the other 3 dimensions. The process dimension embeds more detail of the related process in the states. An example is when the skill 'connect' is specified into 'welding' or the even more specific 'arc-welding' skill. The same applies to the product and resource dimension, they both add more detail to the skill when becoming more product- or resource-dependent.

The concept of states is used to model the impact of an action on the world model. There can be different pre-, post- and hold-states with each it's variables and related conditions. Only the parameters actively used in the post state can be changed by the skill and define the end state. Other parameters can be tracked but will not change permanently because of the skill. In other words, skill states contain the subset of resource parameters with additional conditions important for describing the temporary and definite impact of the skill in the world model.

Specifically in this model, states are modelled by variables and conditions. The variables should always be linked to properties of resources in the world model. If a resource is not yet specified, the root object 'resource' is used. Variable conditions impose constraints on the variable, e.g. end state - position equal to end position. In contrast to variable conditions, skill conditions stay on the

level of skills and lay links and constraints between skills. Skill conditions form a separate model and act as a blackbox within the skill model. Behaviour trees and other constraints related to assistive skills are modellable via skill conditions. By not limiting the number of refinements, the user can go as detailed as necessary. By leaving the interpretation of 'skills' open, different viewpoints can be embedded. The next section further clarifies the model with an example.

3.2   Example application

The concepts of the model are illustrated based on an industrial use case from Flemish Industry. The assembly of a compressor serves as an example, more specifically the insertion of a rotor into the main housing is worked out. Figure 2 gives a detailed description of the skill 'insert', worked out towards operators and filled-in specifically for the compressor assembly.
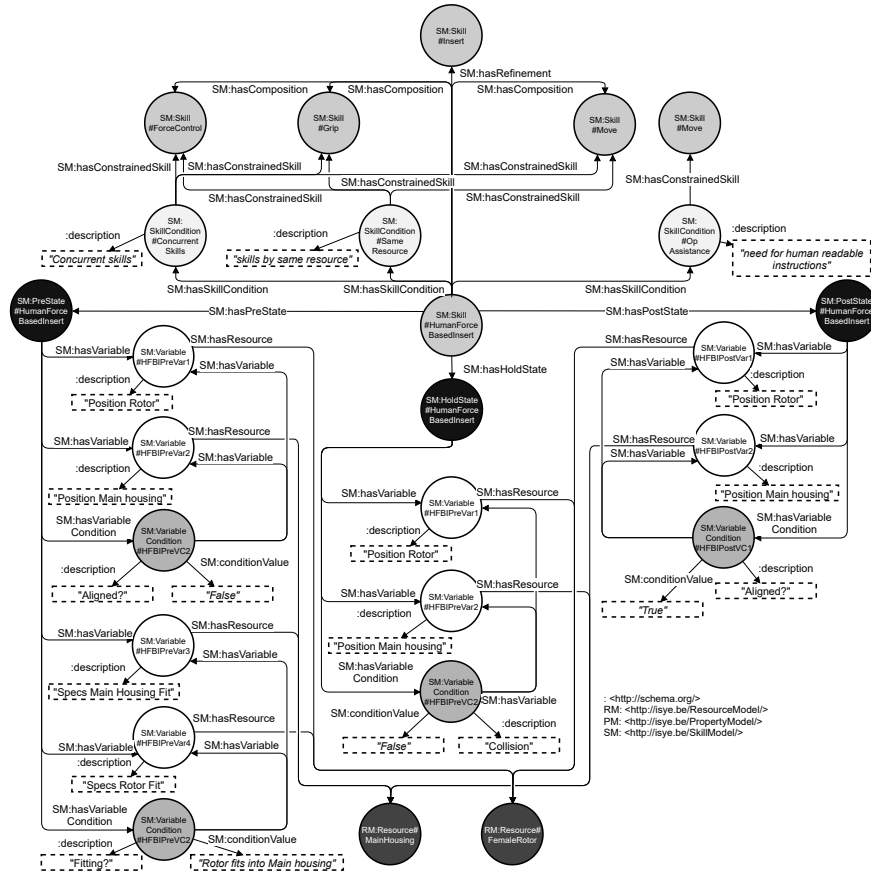

Fig. 2. Skill model example: executable operator insert applied on compressor assembly

During the pre-state the position of both objects that have to be inserted is important together with the shape. Do the objects fit? Are they insertable? The variables directly link to the parts that are manipulated during the insertion. After the skill has taken place, the positions should again be checked and the objects should be aligned. It can be seen that the shape of the objects is important to check if the insertion is possible, but the skill does obviously not change the shape. Therefore 'shape' is not embedded in the post-state. The hold state is used to prevent collisions between the parts during insertion. The model is that detailed that sensors and operator input could be linked to the model to follow up execution. At the same time, in depth information related to assistive skills is available. Because the insert is specified for operators a skill condition is added which imposes the need for a resource which instructs the operator e.g. pick-to-lights, smart glasses etc. Skill conditions can be used to add that kind of assistive skills, but skill conditions can also ask for resources which can store the parts, for tables etc. The skill conditions also impose that the sub-skills should be handled by the same executing resource and be available concurrently. A less refined model of the skill insert can also be added to all resources that have the capacity to insert. This way a process-, resource- and product-independent model will allow easy comparisons between different resources. Can I use a robot, or is an operator more appropriate? The states will then only highlight the more general variables and related conditions. No information on e.g. specific parts or assistive skills will be added. Note that the example is somewhat simplified to keep it printable. Additional states and variables could be added. Nevertheless, this example shows in a comprehensible way the key-strengths of the model.

4   Conclusion and outlook

The design of flexible assembly workstations and optimisation of the execution thereof requires multidimensional and versatile formal models. Skills form the central factor in describing all manufacturing resources and easily link them to production requirements and product flows. In this contribution, a skill-centered formal model is presented which stands out by combining different concepts coming from both the design and execution point of view. The model combines high-level and executable skills and opens the door for assistive skills required to embed, not directly value-adding, peripheral equipment. The model should be further validated, however the applications seem broad and the model seems to easily adapt to different levels of granularity and detail. It is also not obligatory to fill in all details, which is an advantage in contrast to existing models to manage the complexity.

A major drawback of formal models remains however that they always in the first place result in an additional effort. They have to be created next to the real implementations. The strength should come from a broader knowledge store combining all info and facilitating optimisations and new designs in the future. Because of the decoupling of resources, products and processes via a central skill model, data should be more easily and consistently changeable and extendable.

This way experience and knowledge can be consolidated and will not be lost. Firstly, the model will be further validated via additional use cases from industry and adjusted where necessary.

In future work, the task, resource and skill condition model will also be further elaborated. Extra arrangements will be proposed for the definition and meaning of skills, this is key for a neat and well-ordered assembly knowledge store but is nowadays application specific. Nevertheless the formal model will remain the core which opens up compatibility with future plug-ins and to create a shared vocabulary links to standard terminology found in literature and recognized standards will be added.

References

1. Backhaus, J., Reinhart, G.: Digital description of products, processes and resources for task-oriented programming of assembly systems. Journal of Intelligent Manufacturing **28**(8), 1787–1800 (2017). https://doi.org/10.1007/s10845-015-1063-3
2. Ciccozzi, F., Malavolta, I., Selic, B.: Execution of UML models: a systematic review of research and practice. Software and Systems Modeling **18**(3), 2313–2360 (2019). https://doi.org/10.1007/s10270-018-0675-4
3. Danny, P., Ferreira, P., Lohse, N., Guedes, M.: An AutomationML model for plug-and-produce assembly systems. In: Proceedings - 2017 IEEE 15th International Conference on Industrial Informatics, INDIN 2017. pp. 849–854 (2017). https://doi.org/10.1109/INDIN.2017.8104883
4. Hashemi-Petroodi, S.E., Gonnermann, C., Paul, M., Thevenin, S., Dolgui, A., Reinhart, G.: Decision Support System for Joint Product Design and Reconfiguration of Production Systems. IFIP Advances in Information and Communication Technology **566**, 231–238 (2019). https://doi.org/10.1007/978-3-030-30000-5_30
5. Hoang, X.L., Hildebrandt, C., Fay, A.: Product-oriented description of manufacturing resource skills. In: IFAC-PapersOnLine. vol. 51, pp. 90–95. Elsevier B.V. (2018). https://doi.org/10.1016/j.ifacol.2018.08.240
6. Holmström, P.: Modelling manufacturing systems capability. Ph.D. thesis, Kungliga Tekniska Högskolan (2006)
7. Imran, M., Young, B.: The application of common logic based formal ontologies to assembly knowledge sharing. Journal of Intelligent Manufacturing **26**(1), 139–158 (2015). https://doi.org/10.1007/s10845-013-0768-4
8. Järvenpää, E.: Capability-based Adaptation of Production Systems in a Changing Environment (2012)
9. Järvenpää, E., Siltala, N., Hylli, O., Lanz, M.: The development of an ontology for describing the capabilities of manufacturing resources. Journal of Intelligent Manufacturing **30**(2), 959–978 (2019). https://doi.org/10.1007/s10845-018-1427-6
10. Katiraee, N., Calzavara, M., Finco, S., Battini, D., Battaïa, O.: Consideration of workers' differences in production systems modelling and design: State of the art and directions for future research. International Journal of Production Research **59**(11), 3237–3268 (2021). https://doi.org/10.1080/00207543.2021.1884766
11. Kocher, A., Hildebrandt, C., Vieira da Silva, L.M., Fay, A.: A Formal Capability and Skill Model for Use in Plug and Produce Scenarios. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1663–1670 (2020). https://doi.org/10.1109/etfa46521.2020.9211874

12. Malakuti, S., Bock, J., Weser, M., Venet, P., Zimmermann, P., Wiegand, M., Grothoff, J., Wagner, C., Bayha, A.: Challenges in Skill-based Engineering of Industrial Automation Systems. In: IEEE International Conference on Emerging Technologies and Factory Automation, ETFA. pp. 67–74 (2018). https://doi.org/10.1109/ETFA.2018.8502635

13. Pfrommer, J., Stogl, D., Aleksandrov, K., Escaida Navarro, S., Hein, B., Beyerer, J.: Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. At-Automatisierungstechnik **63**(10), 790–800 (2015). https://doi.org/10.1515/auto-2014-1157

14. Rovida, F., Krüger, V.: Design and development of a software architecture for autonomous mobile manipulators in industrial environments. In: Proceedings of the IEEE International Conference on Industrial Technology. pp. 3288–3295. No. June (2015). https://doi.org/10.1109/ICIT.2015.7125585

15. Salunkhe, O., Fast-Berglund, A.: Increasing operational flexibility using Industry 4.0 enabling technologies in final assembly. In: 2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC). pp. 1–5 (2020). https://doi.org/10.1109/ICE/ITMC49519.2020.9198630

16. Siltala, N., Järvenpää, E., Lanz, M.: An Executable Capability Concept in Formal Resource Descriptions. In: IFAC-PapersOnLine. vol. 51, pp. 102–107 (2018). https://doi.org/10.1016/j.ifacol.2018.08.242

17. Van De Ginste, L., Goos, J., Schamp, M., Claeys, A., Hoedt, S., Bauters, K., Biondi, A., Aghezzaf, E.H., Cottyn, J.: Defining flexibility of assembly workstations through the underlying dimensions and impacting drivers. In: Procedia Manufacturing. vol. 39, pp. 974–982. Elsevier B.V. (2019). https://doi.org/10.1016/j.promfg.2020.01.391

18. Vichare, P., Nassehi, A., Kumar, S., Newman, S.T.: A Unified Manufacturing Resource Model for representing CNC machining systems. Robotics and Computer-Integrated Manufacturing **25**(6), 999–1007 (2009). https://doi.org/10.1016/j.rcim.2009.04.014