Resource Interface Matchmaking as a Part of Automatic Capability Matchmaking

Niko Siltala(✉), Eeva Järvenpää, and Minna Lanz

Tampere University, Korkeakoulunkatu 6, 33014 Tampere, Finland
{niko.siltala,eeva.jarvenpaa,minna.lanz}@tuni.fi

Abstract. This paper presents a case study for capability matchmaking method and specifically focuses on interface matchmaking process. This method can be utilised during production system design or reconfiguration by system integrators or end users. They will benefit from fast and automatic resource searches over large resource catalogues. The paper binds together the process around the capability and interface matchmaking, which are presented more in detail in our earlier publications. In this paper, the use of matchmaking process is exemplified, and a verification of the method is provided with two test cases.

Keywords: Interface matchmaking · Assembly system design · System reconfiguration

1 Introduction

Responsiveness of manufacturing is an important strategic goal for manufacturing companies operating in a highly dynamic environment characterized by constant change. Such responsiveness and adaptivity is related to the need to reconfigure and adjust the production and corresponding production system as efficiently as possible to the required changes in processing functions, production capacity, and the dispatching of the orders. [1, 2] To do this, the production system needs an inherent ability to facilitate continual and timely change in its structure and in its functional operations.

Traditionally, the production system design and reconfiguration has been time-consuming process done by the human designer. It includes search for suitable and connectable resources, and it relies heavily on the expertise and tacit knowledge of the system integrators and the end users of the system [3]. Meeting the requirements of fast adaptation calls for new methods and solutions that would drastically reduce the time and effort put into system design [2, 4], both in brownfield and greenfield scenarios. Plug and play interfaces, modern information and communication technologies, formal information models representing resources and products, as well as simulations and other computer-aided intelligent planning tools can all contribute to such methods and solutions [2, 4]. During the system design and re-configuration, new structural configurations are built to fulfil the functional requirements set by the product [4]. Similar to the design of modular products [5] – consideration of interfaces plays an important role in enabling the interchangeability and independence of resource elements. Thus, in order

© The Author(s) 2021
S. Ratchev (Ed.): IPAS 2020, IFIP AICT 620, pp. 51–62, 2021.
https://doi.org/10.1007/978-3-030-72632-4_4

to achieve a feasible structural configuration, the combined production resources must have compatible interfaces.

Within the past two decades, there have been multiple different projects and research [6–8] trying to provide computerized support for system design and reconfiguration planning process. According to [6], the modular architecture paradigm for new production systems, which focuses on the clear functional decoupling of equipment module functionalities and the use of standardized interfaces to promote interchangeability, presents the possibility for developing automated system design and reconfiguration methods. Important steps towards modular assembly equipment and standardized hardware and control interfaces was made, for example, in EU-funded project EUPASS [7]. The recently finished project ReCaM [8], which results this paper also reports, aimed to develop a set of integrated tools for rapid and autonomous reconfiguration of production systems. The approach relies on a formal unified functional description of resources [9], providing a foundation for rapid creation of new system configurations through capability-based matchmaking of product requirements and resource offerings [10].

This paper will present, through two case studies, how the capability matchmaking process operates. Emphasis is placed on the interface matchmaking. First objective is to present how the interface concept for production resources can be utilized as a part of the capability matchmaking [11] procedure. Secondly, we will present, through two cases, how the capability and interface matchmaking work in practise and how they sort out feasible resource combinations for the system designer.

The paper is organized as follows. Section 2 introduces shortly our overall capability matchmaking approach and its associated concepts and data models. Section 3 focuses on the interface matchmaking process in general. In Sect. 4 we present with two cases how the matchmaking process advances and provides the matching resource combinations. Section 5 discusses how we have verified the automatically gained matchmaking results. The specific focus is placed on interface matching viewpoint. In the Sect. 6 we conclude the work done.

2 Capability Matchmaking Process

2.1 Information Models Involved

The capability matchmaking relies on different information models that are used to describe the resources and products in a formalized way. For each different resource, a resource description (RD) file (in XML format) is created and published by the resource provider. These files are saved to different catalogues, either global or local, from where the potential users, in this case the capability matchmaking software, can then utilize the descriptions. The resource description represents the basic characteristics, capabilities, interfaces, and properties of the resource. It can contain links to documentation, CAD models and to illustrative figures. One essential part in the resource description is the capability description of the resource, which selects the relevant capabilities and their parameters from the Capability Model.

Central information model for the matchmaking is the Capability Model. The Capability Model is a formal ontology model which is used to model capabilities and their relations. Capability consists of concept name, such as “Drilling”, “Milling”, “Moving” and so on, and parameters, such as “payload”, “speed” or “torque”. The model consists of simple and combined capabilities, where combined capabilities are combinations of simple capabilities. For instance “Picking” is a combined capability which requires as an input “Moving”, and some sort of “Grasping” capability. The capability model forms a capability catalogue, which is a pool of capabilities that can be assigned to resources to describe their functionality. [12]

The Resource Model Ontology imports the Capability Model and is used to describe manufacturing resources and their capabilities. An instance ontology of Resource Model can be automatically created and populated by reading in the information from RDs. The Resource Model can also be used to model systems composed of multiple resources. Based on the relationships between simple and combined capabilities, it is possible to identify potential resource combinations that can have a certain combined capability. [12]

The Resource Model imports another ontology, called Resource Interface Model, which describes the resource interfaces in detail. It is used during the matchmaking for checking the compatibility of the resources from interface perspective. In other words, it is utilized to assess if two resources can be connected physically together. It includes information about interface definition, including identifier, name, category, gender and purpose of the interface; Interface standard and its characteristics and the standardisation organization; Interface port implementation. [13]

Also, the product requirements are described using formal Product Model ontology. The Product Requirement Description (PRD) contains information about the parts, how they are connected to assemblies and what kind of processes and process parameters should be used. The Product Requirement Description uses the same concepts that are involved in the Capability Model. Thus, there is a link between the resource capabilities and process requirements of the product. [11]

2.2 Overview of Capability Matchmaking Process

Figure 1 represents the capability matchmaking process. The Capability Matchmaking is implemented as a service [14] that the System Designer may call through his desired planning system. When the designer wants to launch this service, he will have to give it certain inputs, for example the matchmaking search space. This input includes description of the resources that should be taken into consideration during the matchmaking, and description of the product (i.e. the PRD), for which a match will be looked for. Thus, the resources and product requirements need to be formally described before the capability matchmaking process can be initialised. These resources in specific search space are automatically processed and read in from RDs to form a Resource Model instance ontology. The input related to the resources may be either a description of the existing system layout (i.e. a resource hierarchy building a system layout) or pool of resources selected from one or various resource catalogues. In the latter case, the input is in form of a flat list of resources creating a resource pool. The Capability Matchmaking software will then process these inputs and provide the matchmaking result to the designer. The result includes information about the resources and resource combinations matching for each process step of the product requirement. This information can then be further processed in the system designer's planning tool where the desired resources are selected according to some company or situation specific criteria. [14, 15]

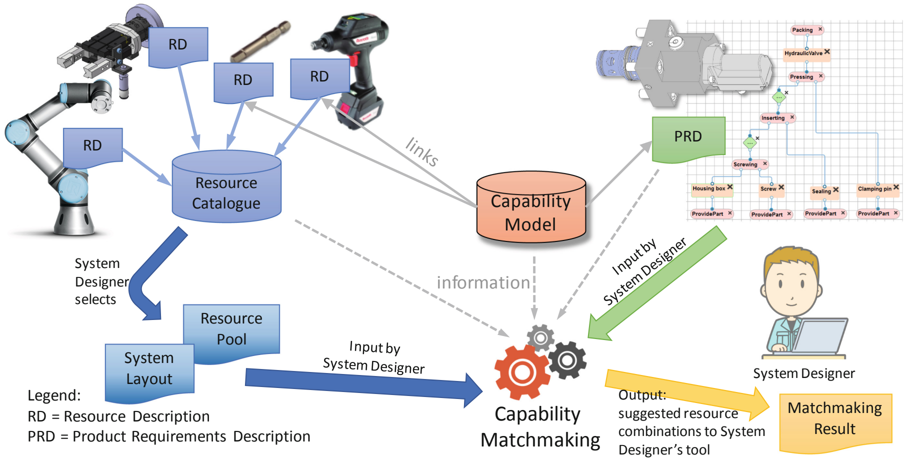

Fig. 1. Capability matchmaking process inputs and outputs

The capability matchmaking involves two aspects: Generation of new resource combinations and matching the capabilities of these combinations with the product requirements. The process flow is illustrated in Fig. 2 from left to right. First, the matchmaking system generates new resource combinations that have capabilities requested by the product, e.g. "Screwing". Next, the interface compatibility of the resources is checked based on the interface matchmaking rules [13, 16]. After that, the combined capability parameters are calculated for the remaining resource combinations based on the combined capability rules. Finally, when the resource combinations have been created and their combined capabilities have been calculated, these combined capabilities are compared to the characteristics and requirements of the product. For this purpose, capability matchmaking rules are used, which find out which resource combinations answer the requirements of the product. The combined capability rules have been introduced in [17] and matchmaking rules in [11]. Both have been implemented with SPIN rule language (SPARQL Inferencing Language).

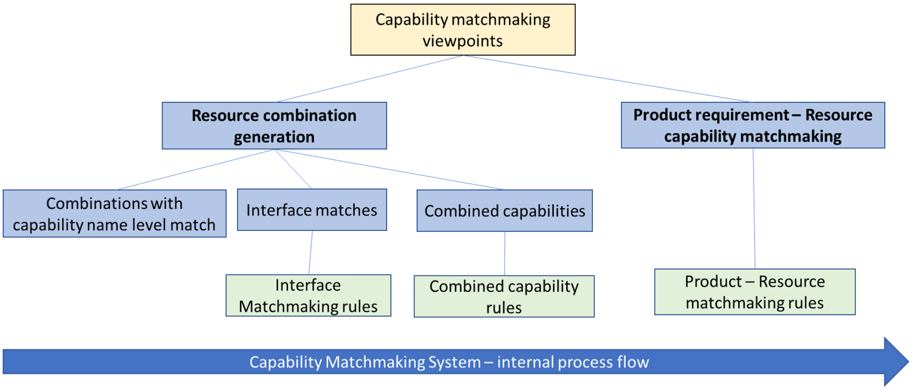
Capability matchmaking viewpoints
Resource combination generation
Product requirement – Resource capability matchmaking
Combinations with capability name level match
Interface matches
Combined capabilities
Interface Matchmaking rules
Combined capability rules
Product – Resource matchmaking rules
Capability Matchmaking System – internal process flow

Fig. 2. Internal process phases in capability matchmaking

3 Interface Matching Process

The objective of the interface matching is to evaluate if a finite set of resources is connectable all together. The interface matching process and algorithm has been explained in detail in [16], and data models, namely Resource Interface Model, and rules guiding the interface matching process in [13]. The interface matching process is independent evaluation procedure working as one phase of capability matchmaking. Once capability matchmaking finds a potential combination of resources, it calls the interface matchmaking procedure with the found resources as a set of tested resources. With the call capability matchmaking provides populated Resource Model ontology as argument. At the end of the call, the interface matching procedure responses with simple true or false response, if the given set of resources are connectable from interface point of view or not.

The interface matchmaking algorithm has two main phases – coarse level and fine level interface matching. The former focuses on the interface code and the gender. The first checks only that the string label representing the coded name of the interface standard is the same at both resource sides. The second defines the polarity of the interface and evaluates it with simple rule – male and female or two neutrals can be connected. Only the coarsely connected resources can continue to fine level interface matching. It utilises the detailed interface characteristics. Each interface contains zero to many additional characteristics. These are properties with name, value and operator, which describes a rule how this characteristic should be compared with a counterpart in order to have successful connection between the resources. Each characteristic is defined as optional or mandatory. All mandatory ones must match with counterpart before resources are deduced connectable at fine level. If the interface does not have any additional characteristics defined, the coarse level match is directly valid at fine level.

The algorithm is implemented with Java. It makes several SPARQL queries to Resource Model and Resource Interface Model ontologies, in order to collect information and reason out connectable resources. It also uses internal data structures, where buffered information and intermediate results are stored for later use. At the end of both interface matching levels, algorithm uses intermediate results to create network out of tested resources. If and only if a single connected network containing all resources is created, it is determined that these resources are connectable. Detailed description of the algorithm is provided in [16]. The software module implementing the interface matchmaking process is now integrated as integral part of overall capability matchmaking process.

4 Case Examples

Two case examples are used to illustrate the capability matchmaking process, specifically focusing more deeply into the interface matchmaking phase. The first case focuses on screw driving and the second on pick and place operations. In practice, capability matchmaking processes all product requirements i.e. each process step one by one and provides output results for each process step at once. For illustration and simpler representation, the cases presented here focus only on a single process step and its requirements and results.

4.1 Screwdriving Solution

This case focuses on a single process step, which defines the requirement for fastening two parts with a M6 sized screw with socket head cap (socket size 5 mm). The required end torque is 13..17 Nm. The matchmaking for this process step is illustrated in Fig. 3, which shows only a subset of our complete test data. First, the matchmaking will create new resource combinations from the available resource pool for the required capability at capability concept name level, i.e. resource combinations that could provide “Screwing” capability. The available resources in the resource pool are illustrated in the up left corner in the figure, while the first phase presents four different combinations created by the matchmaking software. While creating these combinations, matchmaking software will simultaneously check that the interfaces of the resources match at fine level, and that the resources can be physically connected. In case of the resource combination (b) on the second phase, the tool bit does not fit into the screw driver’s tool interface and thus incompatible combination from interface perspective is filtered out.

After that, the matchmaking system will calculate the combined capability parameters for the remaining combinations and check that the capability parameters match with the parametric requirements of the product. During the combined capability calculation, the matchmaking system considers each individual resource and calculates the viable range for the whole combination. I.e. if a tool bit can bear only a certain torque, it can be limiting the overall torque for the whole combination. The combined capability SPIN rules are used for inferring the combination parameters and matchmaking SPIN rules are used to compare the parametric requirements of the product with the capability parameters. In this example case the matchmaking system will check that the screw type and screw head size matches with the tool size and that the provided torque is within the range of the required torque. Unsuitable resources and resource combinations are again filtered out. In this case the combination possibility (a) (in phase three) does not provide enough torque, as it limits in maximum to 6 Nm, and it is eliminated from this

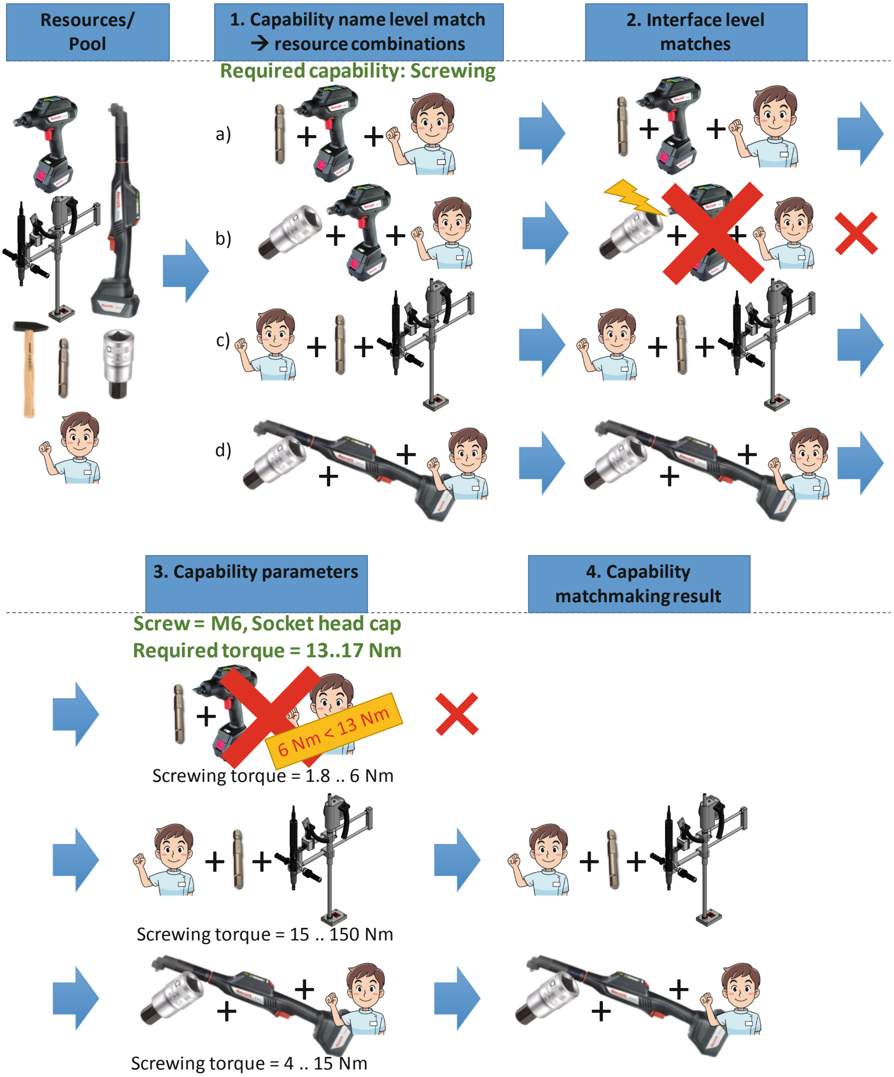
Resources/ Pool
1. Capability name level match
→ resource combinations
2. Interface level matches
Required capability: Screwing
a)
b)
c)
d)
3. Capability parameters
4. Capability matchmaking result
Screw = M6, Socket head cap
Required torque = 13..17 Nm
6 Nm < 13 Nm
Screwing torque = 1.8 .. 6 Nm
Screwing torque = 15 .. 150 Nm
Screwing torque = 4 .. 15 Nm

Fig. 3. Capability matchmaking process for screwing case

result. In the end, only the two suitable resource combinations are left as suggestions in the matchmaking result.

Within our practical test data, the resource pool used for this specific case had 68 resources. 530 resource combinations providing "Screwing" capability where found, and 36 out of those was found compatible from the interface point of view. Combined capabilities were calculated for these and finally two resource combinations were found feasible for the given product requirement. These were presented in Fig. 3.

Table 1 shows interface details for the resources used in this case example. Interface port is a placeholder for interface(s) used for a specific function. Interface standard identifies the code of the standard used. Gender sets the polarity of the interface implementation. Additional interface characteristics define properties characterising the interface, such as size and shape in this example. Comparison operator defines how a characteristic should be used in the compatibility evaluation. In this case operator is 'SAME_SET', which means that at both sides of the connection, the characteristic value(s) must be the same to allow successful connection.

Table 1. Listing of resource, their interfaces and interface characteristics for screwdriving case.

Resource	Port	Interface standard	Gender	IF characteristics	Operator
1. NXP pistolgrip nutrunner	IF1	ISO_1173:2001	F	C.Shape = HEXAGON	SAME_SET
				C.Size = 1–4	SAME_SET
	IF3	HUMAN.INTERACTION.GENERAL	M	–	
2. NXA right-angle nutrunner	IF1	ISO_1173:2001	M	C.Shape = SQUARE	SAME_SET
				C.Size = 3–8	SAME_SET
	IF3	HUMAN.INTERACTION.GENERAL	M	–	
3. Tightening system 350 - EC304	IF1	ISO_1173:2001	F	C.Shape = HEXAGON	SAME_SET
				C.Size = 7–16	SAME_SET
	IF3	COMP-A_MOUNT_0011	M	–	
	IF5	HUMAN.INTERACTION.GENERAL	M	–	
4. Tool bit for screwing - 1/4″ - HexSocket 4 mm	IF1	ISO_1173:2001	M	C.Shape = HEXAGON	SAME_SET
				C.Size = 1–4	SAME_SET
	IF2	ISO_4762:2004	M	C.NominalKeySize = 4	SAME_SET
5. Tool bit for screwing - 1/4″ - HexSocket 5 mm	IF1	ISO_1173:2001	M	C.Shape = HEXAGON	SAME_SET
				C.Size = 1–4	SAME_SET
	IF2	ISO_4762:2004	M	C.NominalKeySize = 5	SAME_SET
6. Tool bit for screwing - 3/8″ - HexSocket 5 mm	IF1	ISO_1173:2001	F	C.Shape = SQUARE	SAME_SET
				C.Size = 3–8	SAME_SET
	IF2	ISO_4762:2004	M	C.NominalKeySize = 5	SAME_SET
7. Tool bit for screwing - 7/16″ - HexSocket 5 mm	IF1	ISO_1173:2001	M	C.Shape = HEXAGON	SAME_SET
				C.Size = 7–16	SAME_SET
	IF2	ISO_4762:2004	M	C.NominalKeySize = 5	SAME_SET
8. Hammer	IF1	HUMAN.INTERACTION.GENERAL	M	–	
9. Human operator - expert	IF1	HUMAN.INTERACTION.GENERAL	F	–	

In Fig. 3. we have the first resource combination (a) from resources 1, 5 and 9. The interfaces allows the combination (See Table 1) – driver (1) has fitting interface (IF1) with the tool bit (5/IF1) at standard, gender and all interface characteristics. Driver (1) has also fitting interface (IF2) with the operator (9/IF1) moving the driver. But later, the combination (a) fails to fulfil the capability parameter requirement for generating enough output torque. The second resource combination (1, 6 and 9) fails at interface matching. The gender of driver (1/IF1) and tool bit (6/IF1) does not match, and even it would, the interface characteristics are not fitting (See Table 1). This would be the case if driver (1/IF1) is tried to connect with tool bit (7/IF1), when the interface characteristic C.Size will prevent the connection. In case of the resource combinations (c) and (d) both interface and capability parameter requirement are matching, leading to propose resource combinations (3, 7, 9) and (2, 6, 9) as a match and potential solution for this product requirement.

4.2 Pick and Place Solution

Another case example focuses on pick and place case. Similar kind of case was discussed as test case in [13]. The handled part is cylinder shaped having diameter of 44 mm and length of 30 mm and it weights 50 g. The product requirement defines following needs for the process step: grasping is done externally, maximum allowed compressive force is 10 N, transportation is needed in XYZ-directions (linear movements), and positioning accuracy is 0.2 mm.

Within our practical test data, the same resource pool with 68 resources is used as in the previous case. It contains two grippes and seven moving devices. The capability name level matchmaking found 12 potential combinations providing "pick and place" capability, and four out of those were found compatible also from the interface point of view. Combined capability parameters were calculated for these and finally three resource combinations were found feasible for the given product requirement. In this case three different robot arms are found mating with one of the grippers. Resource combination (a) from resources 4 and 1; (b) from 5 and 1; and (c) from 6 and 1 (See Table 2). The resource (4) demonstrates use of several standard interfaces for same interface port. In this case the Schunk-SWS interface is implemented with help of an adapter plate, thus a tool fulfilling either of these standards can be connected to this interface port. In case of resource combination (c) the gripper has two interface ports where the connection can take place – 6/IF2 and 6/IF3. The current version of interface matchmaking does not count or reserve the interface ports, but match is returned if there is any suitable position for connection. The fourth resource combination (d) is a manipulator (3) mating with the gripper (2) (See Table 2), but it provides motion only in two degrees of freedom, thus getting neglected from the result.

5 Verification of Capability and Interface Matchmaking Results

Important part of our SW development process was testing and verification. We used automatic unit tests for our SW development, but the verification of the end results was done mainly manually. We defined manually the expected resource combinations, and inferred and calculated the combined capability parameters from the input resource pool and the product requirement in question. This defined a target result, which was then compared with the result of automatic capability matchmaking process. The verification contained mainly two evaluation criteria: a) all resource combinations/resources compared to manually deduced matches are found, and b) no any false positive matches are included. I.e. resource combination is included to the result, even it is not solving the product requirement from whatever perspective or its resources are not connectable together.

In the development phase, we verified separately and gradually the operation of the different phases of our capability matchmaking process. I.e. starting from name level matchmaking and ensuring gradually, that correct sub-set or sub-result was passed on

Table 2. Listing of resources, their interfaces and interface characteristics for pick and place.

Resource	Port	Interface standard	Gender	IF characteristics	Operator
1. Gripper1.1	IF1	Schunk_SWS	F	C.Size = 005	SAME_SET
				C.Pneumatic = Pxx	SAME_SET
	IF3	General_Tool_TCP	M	–	
2. Gripper-disc1	IF1	COMP-A_GRIPPER_0001	M	C.Size = 30	SAME_SET
	IF3	General_Tool_TCP	M	–	
3. Lab_2-axis manipulator	IF1	SHAPE.FLANGE.CIRCULAR	N		
	IF2	COMP-A_GRIPPER_0001	F	C.Size = 30	SAME_SET
4. UR_UR10 ind	IF1	SHAPE.FLANGE.CIRCULAR	N		
	IF2	ISO_9409-1:2004	F	C.Pitch_circle_diameter = 50	SAME_SET
				C.Number_of_thread_holes = 4	SAME_SET
	IF2	Schunk_SWS	M	C.Size = 005	SAME_SET
				C.Pneumatic = Pxx	SAME_SET
5. UR_UR10 collaborative	IF1	SHAPE.FLANGE.CIRCULAR	N		
	IF2	ISO_9409-1:2004	F	C.Pitch_circle_diameter = 50	SAME_SET
				C.Number_of_thread_holes = 4	SAME_SET
	IF2	Schunk_SWS	M	C.Size = 005	SAME_SET
				C.Pneumatic = Pxx	SAME_SET
6. Dual-arm robot	IF1	SHAPE.FLANGE.CIRCULAR	N		
	IF2	Schunk_SWS	M	C.Size = 005	SAME_SET
				C.Pneumatic = Pxx	SAME_SET
	IF3	Schunk_SWS	M	C.Size = 005	SAME_SET
				C.Pneumatic = Pxx	SAME_SET
	IF5	SHAPE FLANGE RECTANGLE 4HOLE	M	P.Hole_pitch_in_X = 0.042	SAME_SET
				P.Hole_pitch_in_Y = 0.042	SAME_SET
7. Balancer device	IF3	General_Tool_TCP	M	–	
	IF4	HUMAN.INTERACTION.GENERAL	M	–	
8. Human operator - expert	IF1	HUMAN.INTERACTION.GENERAL	F	–	

the next phase. This was mandatory because of complexity of our system and eventually timely long run of complete capability matchmaking process.

The verification was comprehensive and truthful, because of the use of representative resources even the amount of resources was limited. We had different kind of resources providing the same simple capability, with good mix of interfaces and capability parameter values. The finite set made it possible to determine and calculate manually all possible resource combinations and their combined parameter values. The processes (capability name matching, SPIN rules, and interface matching algorithm) are expected to operate deterministically, thus the matchmaking should behave similarly and find correct results when the search space (i.e. size of resource pool) is enlarged.

6 Discussion and Conclusions

This paper presented interface matchmaking process and how it can be used to sort out resource combinations as a part of capability matching process. We have integrated the interface matching software module as integral part of the overall matchmaking process, and we have proven that the implementation of the prototype does work as desired. In this paper, the operation of the whole capability matchmaking process was demonstrated through two cases, focusing on interface matching. The presented cases illustrate how the capability matchmaking creates proposed resource combinations, and how interface and parametric matching filter out the non-fitting combinations, leaving eventually only the fitting matches as output. Sub-set of resources, which are involved within the cases, and the resources' interface definitions were described in the paper. The verification of results was discussed. The output of capability matchmaking corresponds with the manually deduced results for the given test resource pool containing 68 resources.

The capability matchmaking method can facilitate rapid system design and reconfiguration planning, by allowing computerised methods to find feasible system configuration scenarios to different product requirements. Use of automatic matchmaking reduces and speeds up the manual design efforts, as the system designer can focus his/her resource selection to truly connectable and fit resources, instead of searching for resources, and analysing their interfaces and properties. The matchmaking can be applied over large resource catalogues containing thousands of resources and their variants, which can be automatically screened to find the few appropriate resources and resource combinations. Additionally, the digital resource catalogues are expected to contain production resources from multiple vendors, which increases number of resources to study, but also increases the number of available alternatives. Thus, matchmaking opens possibilities for new and more innovative solutions to be found. The system designer is not bound to comfortable "old and known solution", which is almost solving the requirements, but he/she can select the optimum one.

As a future work, we have some ideas to continue the interface matchmaking procedure to still finer level of detail including resource matching at interface port level. This can extend the matchmaking procedure for suggesting also physical layouts, not only resource combinations. Testing of capability matchmaking tools and processes will be continued with more resources, and different assembly and manufacturing processes.

References

1. Koren, Y., Shpitalni, M.: Design of reconfigurable manufacturing systems. J. Manuf. Syst. **29**, 130–141 (2010). https://doi.org/10.1016/j.jmsy.2011.01.001
2. Wiendahl, H.-P., et al.: Changeable manufacturing - classification design and operation. CIRP Ann. **56**, 783–809 (2007). https://doi.org/10.1016/j.cirp.2007.10.003
3. Rösiö, C., Säfsten, K.: Reconfigurable production system design – theoretical and practical challenges. J. Manuf. Technol. Manage. **24**, 998–1018 (2013). https://doi.org/10.1108/JMTM-02-2012-0021
4. Westkämper, E.: Factory transformability: adapting the structures of manufacturing. In: Reconfigurable Manufacturing Systems and Transformable Factories, pp. 371–381. Springer, Berlin (2006). https://doi.org/10.1007/3-540-29397-3_19
5. Pakkanen, J., Juuti, T., Lehtonen, T.: Brownfield process: a method for modular product family development aiming for product configuration. Des. Stud. **45**, 210–241 (2016). https://doi.org/10.1016/j.destud.2016.04.004
6. Ferreira, P., Lohse, N., Ratchev, S.: Multi-agent architecture for reconfiguration of precision modular assembly systems. In: Ratchev, S. (ed.) IPAS 2010. LNCS, vol. 315, pp. 247–254. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-11598-1_29
7. EUPASS Consortium: EUPASS - Evolvable Ultra-Precision Assembly SystemS - Project. EU FP6, GA No 507978. https://cordis.europa.eu/project/rcn/75342/factsheet/en (2006). Accessed 11 Nov 2019

8. ReCaM - Rapid reconfiguration of flexible production systems through capability- based adaptation, auto-configuration, integrated tools for production planning - project. EU Horizon 2020, GA No 680759. http://www.recam-project.eu/ (2017). Accessed 11 Nov 2019
9. Siltala, N., Järvenpää, E., Lanz, M.: Formal information model for representing production resources. In: Nääs, I. et al. (eds.) APMS 2016. IFIP Advances in Information and Communication Technology, vol. 488, pp. 53–60. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51133-7_7
10. Järvenpää, E., Siltala, N., Hylli, O., Lanz, M.: Capability matchmaking procedure to support rapid configuration and re-configuration of production systems. Procedia Manuf. **11**, 1053–1060 (2017). https://doi.org/10.1016/j.promfg.2017.07.216
11. Järvenpää, E., Siltala, N., Hylli, O., Lanz, M.: Product model ontology and its use in capability-based matchmaking. Procedia CIRP **72**, 1094–1099 (2018). https://doi.org/10.1016/j.procir.2018.03.211
12. Järvenpää, E., Siltala, N., Hylli, O., Lanz, M.: The development of an ontology for describing the capabilities of manufacturing resources. J. Intell. Manuf. **30**, 959–978 (2019). https://doi.org/10.1007/s10845-018-1427-6
13. Siltala, N., Järvenpää, E., Lanz, M.: Creating resource combinations based on formally described hardware interfaces. In: Ratchev, S. (ed.) Precision Assembly in the Digital Age. IPAS 2018. IFIP Advances in Information and Communication Technology, vol. 530, pp. 29–39. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05931-6_3
14. Mital, A., Siltala, N., Järvenpää, E., Lanz, M.: Web-based solution to automate capability matchmaking for rapid system design and reconfiguration. Procedia CIRP **81**, 288–293 (2019). https://doi.org/10.1016/j.procir.2019.03.050
15. Järvenpää, E., Siltala, N., Hylli, O., Lanz, M.: Implementation of capability matchmaking software facilitating faster production system design and reconfiguration planning. J. Manuf. Syst. **53**, 261–270 (2019). https://doi.org/10.1016/j.jmsy.2019.10.003
16. Siltala, N., Järvenpää, E., Lanz, M.: A method to evaluate interface compatibility during production system design and reconfiguration. Procedia CIRP **81**, 282–287 (2019). https://doi.org/10.1016/j.procir.2019.03.049
17. Järvenpää, E., Hylli, O., Siltala, N., Lanz, M.: Utilizing SPIN rules to infer the parameters for combined capabilities of aggregated manufacturing resources. IFAC-PapersOnLine **51**, 84–89 (2018). https://doi.org/10.1016/j.ifacol.2018.08.239

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.