



27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017,
27-30 June 2017, Modena, Italy

Capability matchmaking procedure to support rapid configuration and re-configuration of production systems

Eeva Järvenpää^{a*}, Niko Siltala^a, Otto Hylli^b, Minna Lanz^a

^aLaboratory of Mechanical Engineering and Industrial Systems, Tampere University of Technology, Korkeakoulunkatu 6, 33720 Tampere, Finland

^bLaboratory of Pervasive Computing, Tampere University of Technology, Korkeakoulunkatu 6, 33720 Tampere, Finland

Abstract

Rapid responsiveness in terms of processing functions, production capacity and order dispatching is required from today's production systems. This paper introduces a capability-based matchmaking procedure, which supports rapid configuration and re-configuration of production systems. The approach relies on formal OWL-based descriptions of both product requirements and resource capabilities. The base algorithm for the matchmaking is introduced together with the required rules for combined capability calculation and capability matchmaking. The presented approach supports the system designer and reconfiguration planner by automatically suggesting possible resource alternatives for certain product requirements.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing

Keywords: Capability model; Capability matchmaking; Production system reconfiguration; Adaptive manufacturing; Ontology

1. Introduction

Responsiveness has become a new strategic goal for manufacturing enterprises alongside with quality and costs [1]. There is a need for rapidly responding production systems that can timely adjust to the required changes in

* Corresponding author. Tel.: +358-40-849-0869
E-mail address: eeva.jarvenpaa@tut.fi

processing functions, production capacity, and the dispatching of the orders. In recent years, a lot of effort has been put towards reconfigurable production systems e.g. by multitude of different EU-funded projects, and several standardization activities focusing on unification of mechanical as well as communication and control interfaces. However, in real factories the reconfiguration of production systems is still rare.

Based on the literature around flexible, reconfigurable and adaptive manufacturing (e.g. [2][3][4][5]), it is difficult to completely differentiate these concepts. Flexibility is often referred to as the ability to adapt to different requirements without physical changes to the system, whereas reconfigurability refers to the ability to change system components when new requirements arise. ElMaraghy [3] divided a manufacturing system's reconfigurability into physical and logical reconfiguration, touching on both definitions of flexibility and reconfigurability. In this work, also parametric reconfiguration is included into the definition. The physical reconfiguration means that physical changes are made to the system, such as changing the layout of the system, adding or removing machines or machine elements (e.g. tooling). During the logical reconfiguration, the changes occur only on logical level, e.g. by adapting the process sequence and re-routing or re-scheduling the production. The parametric reconfiguration refers to changing the adjustable machine parameters, such as the speed of a conveyor line, the force of a press or torque of a screwdriver.

In order to bring the dream of rapid responsiveness of the future manufacturing systems into reality, efficient methodologies, tools and information models are needed. These should support planners and engineers in the reconfiguration planning and system integration process, and also to allow logical and parametric reconfiguration to take place autonomously while the system is running. In the context of distributed intelligent systems or reconfigurable manufacturing, ontologies play a key role as they provide a shared, machine-understandable vocabulary for information exchange among dispersed agents [6][7]. The European Commission funded project ReCaM (Rapid Reconfiguration of Flexible Production Systems through Capability-based Adaptation, Autoconfiguration and Integrated Tools for Production Planning) aims to develop a set of integrated tools for rapid and autonomous reconfiguration of production systems [8]. It aims to support all the three previously mentioned types of reconfiguration. The approach is based on intelligent plug-and-produce capable self-describing Mechatronic Objects (MOs), which are able to auto-program and self-adjust to the required task by utilizing parametric capabilities. The ReCaM approach relies on a unified functional description of resources, providing a foundation for rapid creation of new system configurations through capability-based matchmaking of product requirements and resource offerings. The ontological Capability Model and XML-based Resource Description Concept, used to describe these resources, have been presented with details in earlier publications [9] and [10].

This paper will present the initial version of the matchmaking procedure during which the product requirements are being matched against the resource capabilities. This matchmaking approach facilitates the finding of feasible resources and resource combinations for certain product requirements, and comparison of the existing system capabilities against the new requirements to detect any need for reconfiguration. Thus, the approach supports both the reconfiguration and greenfield design scenarios, although the emphasis here is on reconfiguration of the existing system. The base algorithm for matchmaking procedure is presented together with the associated software tools, data models and example rules used for combined capability calculations and matchmaking.

2. Capability-based matchmaking as part of the reconfiguration process

2.1. Information models and software associated to the matchmaking procedure

To support the matchmaking procedure, four distributed ontologies (OWL) have been developed, as shown in Figure 1a. The Process Taxonomy Model defines the hierarchical categorization of different manufacturing processes, e.g. milling is classified as machining and further as material removing process. The matchmaking relies on a common Capability Model [9], which models the generic capabilities as classes and capability parameters as property restrictions of the ontology. Furthermore, it manages the combined capabilities of multiple co-operating resources by modelling associations between the generic capabilities. The capabilities implement the processes defined in the Process Taxonomy. The resource specific capabilities are modelled with a separate Resource Model ontology, in which the resource specific instances of the generic capabilities are created and filled with resource specific parameter values. The Resource Model is also used to model the systems composed of these resources. Based on the capability associations, it is possible to identify what combined capabilities a certain resource or

resource combination has, or which kind of combinations could form a certain (combined) capability. The Product Model is used to model the product characteristics and associated manufacturing requirements in terms of process steps and their required process functionalities. It can be used to construct a Product Requirement Description (PRD), which is needed as an input for the matchmaking procedure.

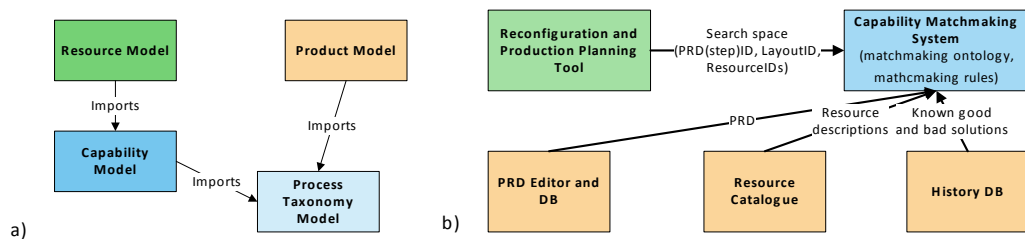


Figure 1. a) Associated OWL-based information models; b) Associated software and databases.

The main software responsible for the capability matchmaking is the Capability Matchmaking System (see Fig. 1b). This system operates with the Capability Matchmaking Ontology, which imports both the Resource Model and Product Model. Integral part of the matchmaking system are the rules, which are used to calculate the combined capabilities of multiple co-operating resources, and matchmaking rules, which are used to compare the PRD with the resource capabilities to find possible matches. The Capability Matchmaking System has access to PRD database and Resource Catalogue(s), from which it retrieves the needed input information for the matchmaking. Furthermore, it can access to History Database in which the history information about previous good and bad combinations and matches are stored. In the reconfiguration planning context, the Capability Matchmaking System will be interfacing with many other software not developed by the authors. For example, the actual reconfiguration planning is done by an external Reconfiguration and Production Planning Tool, which controls the strategy of reconfiguration and thus the search space of the matchmaking. It will be also responsible for selection of resources for the system layout.

2.2. Matchmaking steps

The Capability Matchmaking System will interact with the Reconfiguration and Production Planning tool in an iterative manner. The process and associated information flows relating to the requests and responses between the SWs are shown in the sequence diagram in Figure 2. Due to the space limitations, the sequence diagram is simplified, and it doesn't include all the interactions and SW. For instance, the existing layout cannot be retrieved directly from the resource catalogue and the History DB has been neglected from the diagram.

Search space is the required input information for the matchmaking. It includes the product requirements and available resources that will be used for the matchmaking. Available resources may be e.g. the existing layout and its resources, a list of in-house resources, or a certain catalogue of resources, depending on the matchmaking phase and reconfiguration strategy. The reconfiguration planning tool provides only the resource IDs, while the actual resource information can be retrieved from the resource catalogue(s). The product requirement may refer to the whole PRD of a product, or it may refer to only certain process step(s) in the PRD, depending on the case. The search space will evolve between the matchmaking rounds based on the reconfiguration planner's strategy. The matchmaking procedure could look like this: 1st matchmaking round: check only the current layout; 2nd matchmaking round: allow the current layout to be disassembled and organized in different configurations; 3rd matchmaking round: include in-house resource catalogue to the search space; 4th matchmaking round: include global resource catalogues into the search space; nth round: continue modifying the search space until a feasible match is found to each product requirement step, or even request a modification to the PRD.

The Matchmaking SW returns a list or matrix of the process steps in the PRD and resources or resource combinations matching to these requirements. The system acts only based on the capability information, and it doesn't consider the schedule, price or performance related factors. The selection and optimization of the system is thus to be done by external tools. Furthermore, the matchmaking works with the given resource pool and freely organizes the resources into multiple alternative configurations based on their capabilities. Therefore, the same

individual resource may exist in multiple found matches. It is the task of the reconfiguration planner to decide about which resources and resource combinations to use for each PRD step, and to filter out the results in which the same resource individuals exists. The Reconfiguration and Production Planning Tool has the knowledge on the schedule and constraints imposed to the resources. Thus, it needs to take those into account in the final resource selection, and optimize the system according to the context and situation specific criteria.

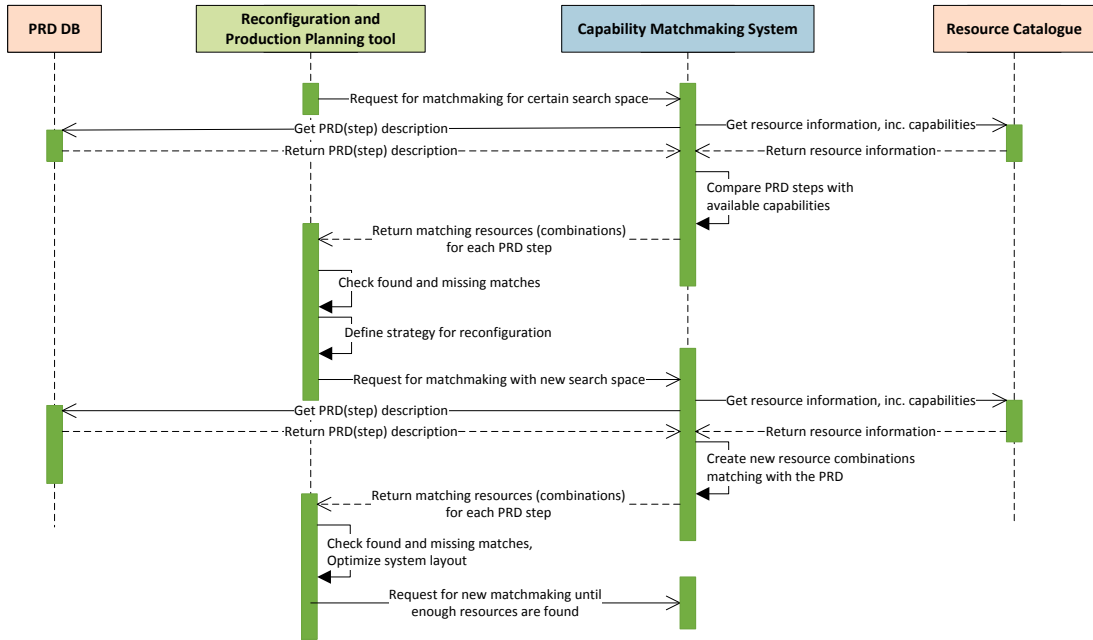


Figure 2. Sequence of the matchmaking procedure and associated SWs.

2.3. Capability matchmaking algorithm

The initial design for internal procedure and logic of the Capability Matchmaking System is presented in Fig. 3. Due to the size and readability of the flowchart, the checking of history data is not included as a separate process, but integrated as part of the steps 2, 3 and 5. In each of these processes, the history data should be checked, if available, after performing the main process, in order to filter out the known unsuitable results.

2.4. Rules needed for matchmaking

When two or more resources are combined together as a functional unit, combined capabilities emerge from the simple capabilities assigned to the individual resources. The parameters for the combined capabilities need to be reasoned out based on the capabilities and properties of the resources involved in the combination. A combined capability is not simply the sum of the individual capabilities it is composed of, but some parameters are directly inherited from the involved simple capabilities, some need to be calculated by a defined formula, some may be overruled or disappeared, and even new parameters may emerge. These combined capabilities and their parameters should be automatically defined and saved to the Resource Model ontology without the need to manually fill in the parameters. For this, combined capability rules are needed. A few example rules of the combined capabilities are given below in a textual form. “Transporting” is a combined capability belonging to resources or resource combinations, which can transport items from one place to another. For instance, conveyor alone or robot combined with a gripper has this capability. In order to automatically define the parameters for the “Transporting” capability

of robot & finger gripper combination, the following rules are applied to give a rough estimate of the combined capability:

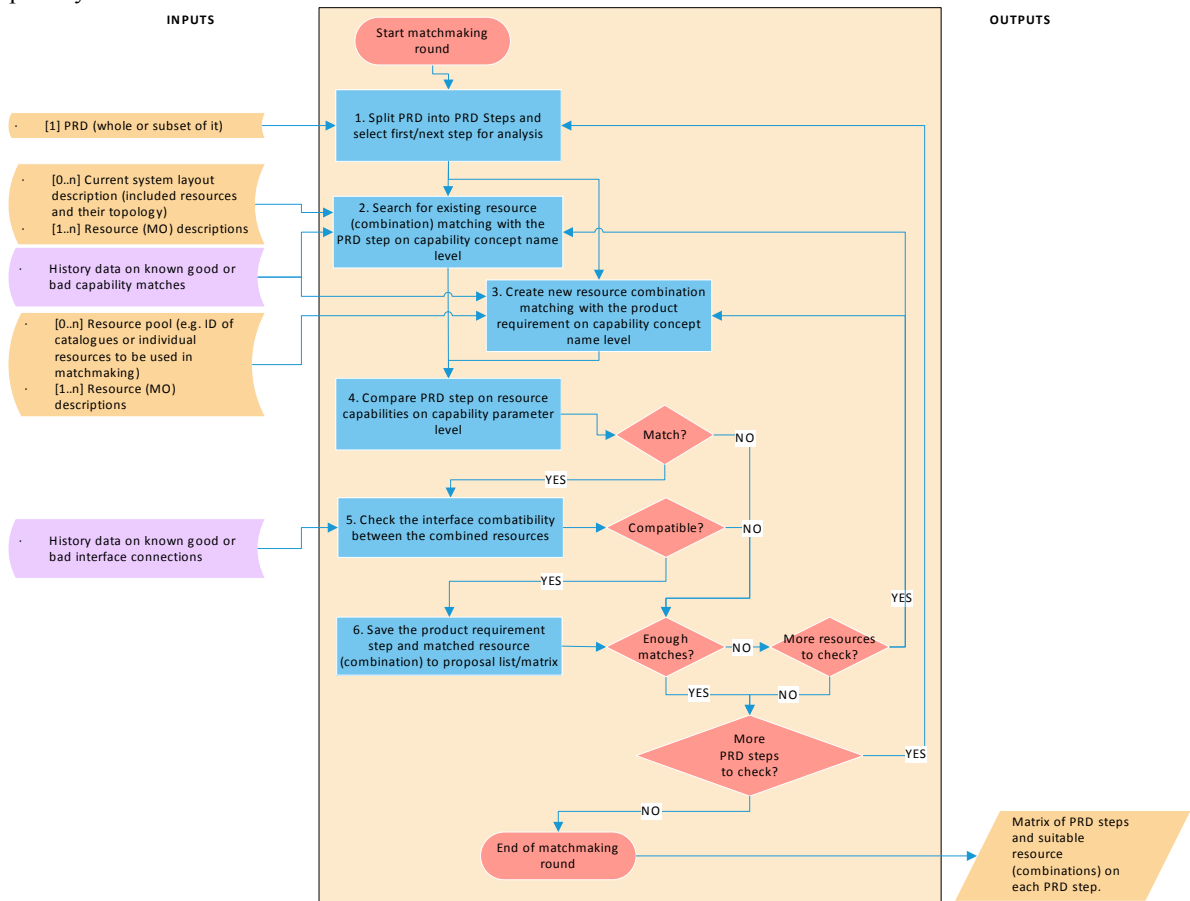


Figure 3. Matchmaking procedure.

- **Transporting *payload*** = “*payload*” property of “Moving” capability subtracted with “*mass*” of the gripper, OR “*payload*” property of “FingerGrasping” capability. The smaller value is dominating.
- **Transporting *itemSize_max*** = “*fingerOpening_max*” property of “FingerGrasping” capability. This applies only to the grasping dimensions, and depends on which way the product is grasped. This needs to be taken into consideration in the capability matchmaking. Similar rule applies for “*itemSize_min*” parameter.
- **Transporting *accuracy*** = “*accuracy*” property of the “Moving” capability. This is only the transportation movement accuracy. In comparison, the *accuracy* property of “Placing” capability = “*accuracy*” property of the “Moving” capability summed up with “*accuracy*” property of “Grasping” capability. This represents the “worst case scenario”.

For implementing these rules, SPARQL (SPARQL Protocol and RDF Query Language) [11] together with SPIN (SPARQL Inferencing Notation) [12] are currently being evaluated and tested. SPARQL queries can be used for querying information from the ontology. SPIN allows rules, written with SPARQL, to be attached to the classes of the ontology in order to increase the reasoning capabilities of pure OWL ontologies. A suitable reasoner tool such as SPIN api can then infer this extra information created by the rules and use it for example in SPARQL query execution.

```

a)
SELECT ?graspingDevice ?movingDevice ?combinedPayload
WHERE {
  ?graspingDevice rmt:hasCapability ?fingerGrasping .
  ?fingerGrasping a cm:FingerGrasping .
  ?graspingDevice rmt:hasBasicResourceInformation ?info .
  ?info cm:mass ?mass .
  ?fingerGrasping cm:payload ?gripperPayload .
  ?movingDevice rmt:hasCapability ?moving .
  ?moving a cm:Moving .
  ?moving cm:payload ?movingPayload .
  BIND ((?movingPayload - ?mass) as ?alternative) .
  bind ( if( ?alternative > ?gripperPayload, ?gripperPayload, ?alternative)
        as ?combinedPayload)
}

b)
CONSTRUCT {
  ?this cm:payload ?payload
}
WHERE {
  BIND ( :getPartCapability( ?this, cm:FingerGrasping ) AS ?fingerGrasping ) .
  FILTER ( bound( ?fingerGrasping ) )
  ?graspingDevice rmt:hasCapability ?fingerGrasping .
  ?graspingDevice rmt:hasBasicResourceInformation ?info .
  ?info cm:mass ?mass .
  ?fingerGrasping cm:payload ?gripperPayload .
  BIND ( :getPartCapability( ?this, cm:Moving ) AS ?moving ) .
  FILTER ( bound( ?moving ) )
  ?moving cm:payload ?movingPayload .
  BIND ((?movingPayload - ?mass) as ?alternative) .
  BIND ( IF ( ?alternative > ?gripperPayload, ?gripperPayload, ?alternative) as ?payload )
}

```

Figure 4. a) Example SPARQL query for calculating payload of Transporting capability; b) Example of SPIN rule for calculating the payload of Transporting capability.

The combined capability rules are required in two scenarios: 1) Finding the parameter values for the capabilities of an existing resource combination; 2) Searching for a suitable combination of resources for a specific capability requirement. In the second case, it is possible to use SPARQL queries to find the required information. Figure 4a shows an example of SPARQL query, which finds all possible resource combinations having combined capability “Transporting” (formed from “Moving” and “FingerGrasping” capabilities) and shows the combined “*payload*” parameter. In order to define all the combined capability parameters for a certain resource combination, it is more convenient to utilize SPIN rules integrated into the ontology file, instead of separate SPARQL query for each parameter. Figure 4b shows a SPIN rule for calculating the payload of transporting capability. This rule utilizes another feature of SPIN, namely SPIN functions, where useful SPARQL queries can be turned in to functions, which then can be used in other queries.

Another set of rules, which relate to the combined capabilities, are the interface matching rules. These rules are used to check the interface compatibility between the resources. Such kind of resources, which cannot be (physically) combined together, must be filtered out from the search results. No further details about these rules can yet be given, as the interface part of the Resource Model has not yet been detailed.

After the combined capabilities have been calculated, they can be compared against the Product Requirement Description, in order to find possible matches. Those capabilities, which match to the PRD step on capability concept name level, are now compared on the parameter level. For implementing this, a rule-based approach is again utilised. The specific implementation technologies of capability matching rules have not yet been tested. However, it is assumed that the same technologies can be applied as for combined capability rules, because the capability matching rules are very similar to the ones presented earlier. For example, if product requires drilling a hole with diameter $D = 10$ mm and length $L = 50$ mm, then the rules will be used to check if the provided “Drilling” capability has “*hole_diameter*” parameter value exactly 10 mm, and “*max_drilling_depth*” larger than 50 mm. If product requires “Pick and Place” operation, the rules will be used to check that the “*payload*” of “Picking” and “Transporting” capabilities is bigger than the product’s mass, and that the product size is within the range of the “*allowedItemSize_min/max*” of these capabilities. The capability matching rules handle the capabilities on upper level, e.g. comparing the parameters of Picking, instead of “FingerGrasping”. This is because “Picking” can be implemented also with “VacuumGrasping” capability. Therefore, the matchmaking rules compare the “*allowedItemSize*” parameters, which have already been calculated for the “Picking” capability, rather than “*fingerOpening_max/min*” parameters of the lower level capability “FingerGrasping”. Firstly, the aim is that the PRD does not need to specify exactly, which capability should be used, and secondly that no specific matchmaking rules would need to be defined for each lower level simple capabilities.

3. Example case

In this section, a simplified use case, implemented in the laboratory environment of and industrial partner, is shown to illustrate the presented concepts. The case product is a tower assembly of two discs stacked on each other. The upper part of the figure 5 represents the product requirements, while the lower part represents the existing

system setup and its capabilities. Due to the limited space, only one functional combination of resources, i.e. the “ReCaM Lab manipulator combination”, consisting of 2-axis manipulator and finger gripper, is included into the capability example. The combined capabilities, illustrated on the right side of the figure, are automatically calculated based on the combined capability rules. Capability matchmaking rules are then used to check the product data against the capability parameters.

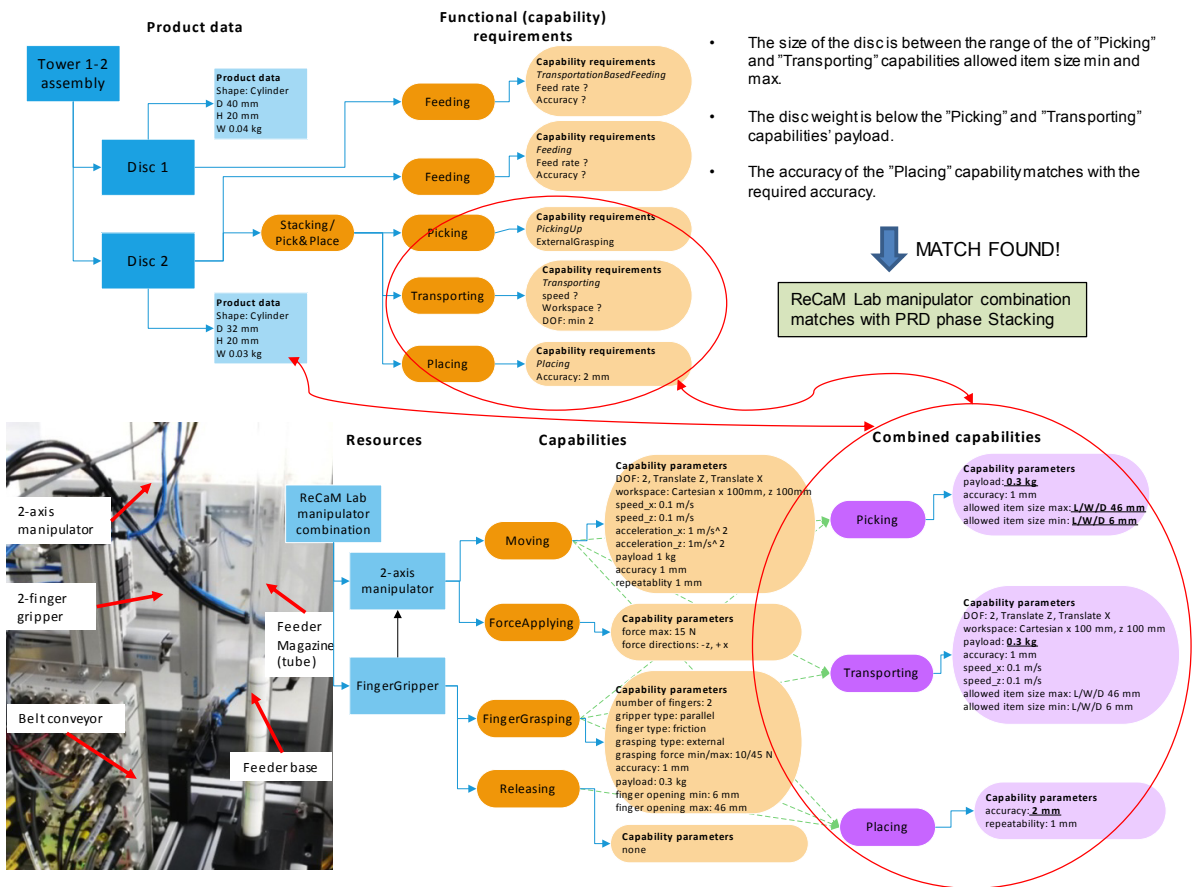


Figure 5. Case example.

The presented approach facilitates reconfiguration planning and reactive adaptation by automatic decision support. The ability to derive the combined capabilities, based on the defined Capability Model and associated rules, eliminates the need to describe the resource combinations manually, but they can be dynamically created based on the individual resource descriptions. In addition, the existing resource combinations can be decomposed into single resources, which allows automatic reasoning methods to provide suggestions for reconfiguration. Such suggestions may relate to e.g. changing the magazine of a tube feeder or gripper of a robot, or changing a certain resource parameter such as torque of a screw driver.

All the parameter examples given in the product requirement side, e.g. “speed” or “feedRate”, are not actually determined by the product itself, but by the performance and process optimization requirements of the system designer. Thus, it may not be feasible to give such parameters in the PRD itself. However, the Matchmaking Software can take such information as an input, as the resource capability descriptions include the information needed for such reasoning. Thus, the input for the Capability Matchmaking Software, may either be pure product requirement description, or it may also include production performance related parameters. Such parameters can be

used, for instance, in later phases of the system design and reconfiguration, when making decisions about the optimal resource selection.

4. Discussion and conclusions

This paper discussed about the capability matchmaking approach, which relies on formal OWL-based representations of product requirements and system capabilities. The presented approach uses combined capability rules to define the capabilities of multiple co-operating resources. It has to be noted that in many cases these rules produce highly simplified and crude results. This is due to the fact that many of the properties of the combined capabilities emerge as a behavior of the machine or station as a whole in a certain context and environment, and they cannot be decomposed into the properties of the various components. Thus, the rules are highly simplified. Furthermore, the aim of the combined capability rules is not to provide detailed analysis of, e.g. the workspace or the kinematics of the device, but to enable the modelling of scenarios for potentially suitable resource combinations. For kinematics and detailed workspace definitions, virtual simulation tools should be used to validate the results obtained from the reasoning based on the digital information.

The presented approach is expected to reduce the workload of a system designer and reconfiguration planner, by automatically filtering out impossible solutions from large search spaces and suggesting possible solution alternatives for the system design and reconfiguration. The presented approach supports at least the following activities: Finding resource combinations that fulfil a given product requirement; Checking if the current system is able to fulfil the given requirement; Making sure that all the needed system components exist in the system; Indicating any missing capabilities in the existing system and suggesting alternative resources having that capability; Showing the parameter range of the available capabilities, facilitating the decision about parametric adaptation; Showing the available capabilities on the factory floor and their coarse location providing necessary information for the logical adaptation (e.g. re-routing); and Automatic configuration scenario generation based on the given requirements. Therefore, it eases and speeds up the system design and reconfiguration planning, and facilitates even the reactive adaptation.

Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 680759.

References

- [1] Koren, Y. & Shpitalni, M., Design of reconfigurable manufacturing systems, *J. of Manufacturing Systems*, 29(4), 2010, pp. 130-141.
- [2] ElMaraghy, H.A. (Ed.), *Changeable and Reconfigurable Manufacturing Systems*, Springer, 2009, ISBN 978-1-84882-066-1.
- [3] ElMaraghy, H. A., Flexible and reconfigurable manufacturing systems paradigms. *Int. J. of Flexible Manufacturing Systems*, Vol. 17, No. 4, 2006, pp. 261-276, ISSN 0920-6299.
- [4] Koren, Y., General RMS Characteristics, Comparison with Dedicated and Flexible Systems, In: *Reconfigurable Manufacturing Systems and Transformable Factories*, A. I. Dashchenko (Ed.), pp. 27-46, Springer, 2006, ISBN 3-540-29391-4.
- [5] Tolio, T. & Valente, A., An Approach to Design the Flexibility Degree in Flexible Manufacturing Systems, *Proceedings of the 16th Int. Conference on Flexible Automation and Intelligent Manufacturing*, 2006, pp. 1229-1236.
- [6] Ameri, F., Urbanovsky, C. & McArthur, C., A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling, *Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing*, 2012, 14 p.
- [7] Leitão, P., Colombo, A. W. & Karnouskos, S., 2016, Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges, *Computers in Industry*, 81, pp. 11–25.
- [8] ReCaM consortium, ReCaM project web page, <http://www.recam-project.eu>, [Accessed 15.8.2016].
- [9] Järvenpää, E., Siltala, N. & Lanz, M., Formal Resource and Capability Descriptions Supporting Rapid Reconfiguration of Assembly Systems, In *Proceedings of the International Symposium on Assembly and Manufacturing*, IEEE, 2016. p. 120-125.
- [10] Siltala, N., Järvenpää, E. & Lanz, M., Formal Information Model for Representing Production Resources. *Proceedings of International Conference on Advances in Production Management Systems*, APMS, 2016.
- [11] W3C, SPARQL Query Language for RDF, 2008, Available in: <http://www.w3.org/TR/rdf-sparql-query/> [Accessed 10.3.2016].
- [12] W3C, SPIN – Overview and motivation, 2011, Available in: <https://www.w3.org/Submission/spin-overview/> [Accessed 15.12.2016].