Capability matchmaking software for rapid production system design and reconfiguration planning

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

a Faculty of Engineering and Natural Sciences, Tampere University, Tampere, Finland
b Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland

A R T I C L E   I N F O

Keywords:
Production system design
Production system reconfiguration
Capability matchmaking
Matchmaking software
Resource modelling
Ontology

A B S T R A C T

Traditionally, the production system design and reconfiguration planning are manual processes, which rely heavily on the designers’ expertise and tacit knowledge to find feasible system configuration solutions. Rapid responsiveness of future production systems calls for new computer-aided intelligent design and planning solutions, that would reduce the time and effort put into system design, both in brownfield and greenfield scenarios. This paper describes the implementation of a capability matchmaking software, which automatizes the matchmaking between product requirements and resource capabilities. The interaction of the matchmaking system with external design and planning tools is explained and illustrated with a case example. The matchmaking approach supports production system design and reconfiguration planning by providing automatic means for checking if the existing system already fulfills the new product requirements, and for finding alternative resources and resource combinations to specific product requirements from large search spaces, e.g. from global resource catalogues.

© 2020 The Author(s). 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/)

1. Introduction

Responsiveness 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 design, reconfigure and adjust the production and corresponding production system as efficiently as possible to the required changes in processing functions, production capacity, and dispatching of the orders. Traditionally, the production system design and reconfiguration planning are manual processes, and heavily dependent on the designers’ expertise and tacit knowledge to find feasible system solutions by comparing the characteristics of the product to the technical properties of the available resources. This process is slow, and it sets limitations to the amount of potential system configuration alternatives that can be considered. Meeting the requirements of fast adaptation calls for new computer-aided intelligent planning and decision support solutions, that would reduce the time and effort put into system design, both in brownfield (reconfiguration) and greenfield (new system design) scenarios.

A key enabler of smart manufacturing is the virtualization of physical assets of the manufacturing, namely resources and products (Lu and Xu, 2018; Tao et al., 2018). There is a need for formal, structured representation of resources and products that allow the resource vendors to describe the functionality of their offerings in a comparable manner, and system designers to make a match between the product requirements and resource capabilities. Formal engineering ontologies and other Semantic Web technologies have become popular solutions for addressing the semantic interoperability issue in heterogeneous distributed environments (Ameri et al., 2012; Jardim-Goncalves et al., 2016; Leitão et al., 2016). Several different semantic ontology-based descriptions have been proposed for the service description in Cloud Manufacturing context (e.g. (Luo et al., 2013; Yuan et al., 2017; Lu and Xu, 2018)). Lu and Xu (2018) presented a service composition and mapping approach based on ontological description of the services and Jena rules to compare the service request with the offering. Manufacturing Service Description Language (MSDL) was developed as a formal domain ontology for representing the capabilities of manufacturing services, focusing on mechanical machining and metal casting services (Ameri et al., 2012), and used for a matchmaking methodology, which aims to connect buyers and sellers of manufacturing services in distributed digital manufacturing environments (Ameri and Patil, 2012). Ameri and McArthur (Ameri and
McArthur, 2014) utilized SWRL (Semantic Web Rule Language) for intelligent supplier discovery based on the services they provide.

Despite there being several approaches to service description and resource virtualization, the current research around service and capability matching has concentrated more on the theory and framework, rather than comprehensive descriptions of the resources and services, and practical implementations of the matchmaking. In addition, the previous research efforts to describe manufacturing resource capabilities have not considered the combined capabilities of multiple co-operating resources, and consequently have not included mechanisms to infer the aggregated parameters of combined capabilities. Thus, the existing approaches cannot provide a solution for capability matchmaking in the context of production system design and reconfiguration planning.

During the EU funded project ReGaM, we have developed computerized support for the system design and reconfiguration planning process. The approach relies on formal description of resources and products, providing a foundation for rapid creation of new system configurations through capability-based matchmaking of product requirements and resource offerings (Järvenpää et al., 2018a, 2018b, 2018c, Siltala et al., 2018a, 2018b). The matchmaking system allows the designer to search through large resource catalogues to find feasible resource combination alternatives without manual effort.

In this paper, we will introduce the implementation of the matchmaking software, and how it can be utilized by external design and planning tools. The paper is organized as follows. In Section 2 we will introduce the capability matchmaking system and its associated concepts, information models, and software architecture. In Section 3 we will explain how external design and planning tools can utilize the matchmaking software through its Web Service interface. In Section 4 we will give an example of the use of the matchmaking system and its internal functionality. Finally, we will end with discussions and conclusions in Section 5.

2. Capability matchmaking system

The matchmaking system intends to ease up the system design and reconfiguration planning procedure by automatically suggesting alternative resource combinations for specific product requirements. In this section we will first discuss about the formal information models used as an input for the matchmaking. Secondly, we will shortly explain the matchmaking procedure relying of rule-based reasoning. Thirdly, we will introduce the implemented software architecture for the matchmaking system.

2.1. Involved information models

The foundation of the capability matchmaking is the formal information models representing the product requirements and available manufacturing resources. We have developed OWL (Web Ontology Language) based information models to represent this information, and introduced them in our earlier publications (Järvenpää et al., 2018a, 2018b, 2018c, Siltala et al., 2018a, 2018b). Ontology Engineering Methodology (Sure et al., 2009) was followed during the model development. The models are available for download in (Järvenpää et al., 2019a).

The Resource Model ontology is used to describe the available manufacturing resources, including their capabilities, interfaces and other characteristics, as well as systems composed of multiple resources. The Resource Model (Järvenpää et al., 2018a) imports two other ontologies, namely Resource Interface Model and Capability Model. The Resource Interface Model (Siltala et al., 2018a) is used to give a formal description of the resource interfaces.

The Capability Model (Järvenpää et al., 2018b) formalizes functionalities of resources and parameters related to these functionalities. It also defines relations between simple (atomic) and combined capabilities. For instance, a robot can have a simple capability “Moving” and a gripper can have simple capabilities “Grasping” and “Releasing”. Together they can create combined capabilities “Pick and Place” and “Transporting”. Based on these formalized relations, the potential resource combinations that have a certain combined capability can be identified programmatically by utilizing information provided by SPARQL queries. The Capability Model imports another ontology called Process Taxonomy Model, which categorizes different manufacturing and assembly processes in a hierarchical structure.

Product Model ontology (Järvenpää et al., 2018c), can be used to describe the product requirements for the matchmaking. The Product Model describes the parts and their basic characteristics, sub-assemblies and their contained parts, processes related to the parts and sub-assemblies, capability requirements related to the processes, and sequence of the processes. The Product Model also imports the Process Taxonomy, which allows to build a link between the product requirements and provided capabilities during the matchmaking.

The matchmaking is performed with Matching ontology, which imports both Resource and Product Model ontologies. It includes also the matchmaking rules discussed in the following section.

2.2. Matching stages and rules

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. 1. 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 (Siltala et al., 2018b). 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. The combined capability rules have been introduced in (Järvenpää et al., 2018a) and matchmaking rules in (Järvenpää et al., 2019a). Both have been implemented with SPIN rule language (SPARQL Inferencing Notation), which is a W3C Member Submission that has become the de-facto industry standard to represent SPARQL rules and constraints on Semantic Web models (SPIN working group 2017). SPIN can be used to link class definitions with SPARQL queries to capture constraints and rules that formalize the expected behavior of those classes.

![Fig. 1. Internal process phases in Capability Matching.](image-url)
suitable reasoner tool such as SPIN API (Knublauch, 2016) can then infer new information created by the rules, and assert it to the ontology for later use e.g. in SPARQL query execution.

2.3. Implementation of the capability matchmaking software

The capability matchmaking software follows the principles of client-server architecture. It is constructed from three main components: the capability matchmaking web service; the software packages for executing the capability matchmaking process; and the formal information models, discussed in the previous section. The web service and associated software modules are deployed and hosted on Apache Tomcat server.

Fig. 2 outlines the layered architecture of the capability matchmaking system, and interactions between the various layers. It also illustrates the technologies and languages used for the software implementation. The Data Model Layer contains the ontology and other data models needed for the matchmaking. The Data Layer represents the actual data, i.e. instances, used during the matchmaking. The Business and Data Access layers run and execute the matchmaking procedures. The top most layer represents different client systems, which interact with the web service component of the system in order to trigger matchmaking or to obtain the matchmaking results.

The Web Service layer is implemented as a RESTful web service. This choice for interface allows easy and loose coupling for client applications to connect with the matchmaking system in order to send requests and receive responses.

The most important packages in the architecture from the matchmaking reasoning perspective are the Capability Query Library (CQL) and the Matchmaker on the Business layer. The Matchmaker is responsible for sequencing and managing the matchmaking process, and performing the actual matchmaking for the incoming requests. It takes care of the execution of the various SPIN rules through the Java-based CQL API (Application Programming Interface). It creates resource combination possibilities, calculates the combined capability parameters for the involved resource combinations, checks the interface compatibility of the resources, executes matchmaking rules from the Matchmaking Ontology and constructs the matchmaking result from the rule inferences. CQL uses the open source Jena semantic web framework (Apache Software Foundation 2017) and Openllet reasoner (Openllet API, 2019) for working with the ontology models. Jena and Openllet themselves do not support SPIN. Thus, another open source library that builds on top of Jena, called SPIN API (Knublauch, 2016), is used to execute these rules.

3. Interaction of the capability matchmaking system with external software

The Capability Matching is implemented as a web service, and there is currently no integrated graphical user interface (GUI) for it. The aimed use case is that the system designer calls the service through his desired planning system, which follows the specified XML message schemas (XSD) or JSON messages for interacting with the web service. Even the matchmaking system itself doesn’t have a GUI, it can be called manually by sending the requests (and receiving results) through the service URL as XML or JSON formatted text.

As an input, the matchmaking system needs the search space to consider. This includes the Product Requirement Description (PRD) and a set of Resource Descriptions (RD) forming the Resource Pool that ought to be considered during the matchmaking process. In case of reconfiguration scenario, also the description of the existing system layout should be included. The inputs are provided to the matchmaking software by the client application in the form of PRD IDs and RD IDs. The search space is then retrieved and read into the Matchmaking Ontology from various catalogues storing the actual information content. For instance, the resource information is collected from Resource Catalogue where resource providers have supplied descriptions of their offerings in the Resource Description format (Siltala et al., 2018a). The PRD is represented with the Product Model ontology format (Järvenpää et al., 2018a).

The Capability Matching system takes the capability requirements and match them with the capabilities existing on the current system (in case of reconfiguration scenario), or create new resource combinations that match with the requirements. As an output, it provides the matchmaking result. It includes the IDs of the resources and resource combinations matching to each process step defined in the PRD.

The matchmaking is a time-consuming process, and therefore the interaction was implemented as asynchronous calls. Fig. 3 shows the interaction between an external application (client) with

![Fig. 2. Overall software architecture of the capability matchmaking system, modified from (Järvenpää et al., 2019b).](image-url)

![Fig. 3. Interaction of the capability matchmaking software with other external tools and databases.](image-url)
the capability matchmaking system to create the matchmaking search space (sequences 1 and 2), to trigger the matchmaking process with a matchmaking request (sequence 3) and to request the matchmaking result (sequence 4).

The matchmaking system provides the found matches to the external design tools, where the information can be further processed. In addition to the resource identification information, various business properties, performance and reliability related information is collected and delivered along the resource record in the matchmaking result. The designer can use the provided information to evaluate the suitability of the suggested resource(s) for a specific task, and to configure the system based on valued optimization criteria, such as availability, performance, least number of reconfigurations, or costs. The designer will then allocate the resources to specific production tasks and workstations. The matchmaking process can be repeated, with different inputs, until all the tasks have a match or other optimization criteria for resource selection is achieved.

The matchmaking result delivered to the client does not consider the allocation of resources for a specific process step or workstation. It deals each process step individually, and thus it is the duty of the client application (or the designer) to consider the optimum resource allocation and layout of the production system. For example, if the designer allocates a resource instance to a specific process step and a physical location (e.g. workstation), he/she cannot take advantage of any later process step in another location, without having more resource instances of the same kind. In this sense, the matchmaking result delivers only potential possibilities from the perspectives of capability parameters and physical interface connectivity, and the client application (or designer) needs to ensure that the production system can actually be built from the available resources.

During the ReCaM project, interaction of the matchmaking system with two client applications was tested. These were the Flexible System Engineering Platform meant for greenfield system design, and Integrated Reconfiguration and Production Planning tool intended for brownfield system design (Colledani et al., 2018). See (Järvenpää et al., 2019b) for more details of the interaction testing.

4. Matching case example

This section will explain the usage and internal functionality of the matchmaking system by an example scenario. In the scenario, the production system designer designs a new production system for a switch valve assembly process and utilizes the capability matchmaking system to find feasible resources and/or resource combinations. The figures show real test data from running the matchmaking system through its web service interface.

Fig. 4a demonstrates the inputs provided by and outputs received by the client system. The left side of Fig. 4a illustrates the matchmaking request and right side the matchmaking result. Firstly, the request contains a reference to the PRD of the switch valve product. This PRD contains all the process steps and their requirements included in the valve assembly process, but for illustration, we focus here on one process step “stick sub-assembly screwing” (green arrow). It requires screwing a M6 hexagon socket head screw to the end torque between 13 and 17 Nm. The PRD is graphically illustrated in the top left corner of Fig. 4a. The lower left corner illustrates the actual request message sent through the web service interface in XML-format.

Secondly, the request contains reference to the resource search space. In brownfield case it would be an existing system layout, but in case the designer is working with greenfield case, and such production system does not exist. Thus, he/she selects some production resource catalogue(s) and specific resources (if not all), and creates a Resource Pool out of them (Fig. 3/ Sequence 1). The reference to the selected resource pool is illustrated by blue arrow in Fig. 4a.

After the request has been sent through the designer’s client system, the matchmaking is performed for each process step one by one. Fig 4b represents the internal reasoning process of the matchmaking system for dealing with the specific process step requiring “Screwing” capability. Only a subset of our complete test data is shown. First, the matchmaking creates 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 left in the figure, while the first phase presents four different combinations created by the matchmaking software (i-iv). While creating these combinations, matchmaking software will simultaneously check that the interfaces of the resources are compatible, and the resources can be physically connected. In case of the resource combination (ii) on the second phase, the tool bit does not fit into the screw driver’s tool interface and thus the 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 will limit 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 type and size and that the required torque is within the range of the provided torque. Unsuitable resources and resource combinations are again filtered out. In this case the combination possibility (i) (in phase three) does not provide enough torque, and it is eliminated from this result. In the end, only two resource combinations are left as feasible suggestions in the matchmaking result. The alternative (iv) is the same that is visualized as a result in Fig. 4a.

5. Discussion and conclusions

We presented the implementation of a capability matchmaking software and its interaction with external design and planning tools through its web service interface. This service can be exploited both during greenfield and brownfield system design, to automatize the search for suitable resources and resource combinations for specific product requirements. The matchmaking system utilizes formal OWL-based information representations of both products and resources, and SPIN rules to infer new knowledge from those representations. The matchmaking service and software take inputs from different client systems, executes the various rules needed during the matchmaking process, and delivers the results back to the clients.

The developed approach contributes to the existing resource virtualization and matchmaking research by providing means for modelling and reasoning about the combined capabilities of multiple co-operating resources. It describes the parameters relating to the capabilities, and utilizes SPIN rules to infer the parameters of combined capabilities from the parameters of the simple capabilities and to insert them to the model. It is a unique approach and implementation, which has not been presented by other researchers. Similar conceptual ideas for capability matching have been presented, e.g. in (Ameri and Patil, 2012). The main difference, however, lies in the ability to automatically manage
combined capabilities. First of all, it allows the resources to be described on a lower level of granularity, and secondly, it eliminates the need to describe the combined capabilities manually for each possible resource combination. The resource combinations can be dynamically created for certain requirements based on the descriptions of the single resources.

The capability matching reasoning actions, discussed in this paper, can be performed automatically based on the defined SPIN rules. However, the results should be validated by human designer during the layout design and resource selection, as the information models and rules are always a simplified representation of the real world. Some of the capabilities depend on the physical location between the combined resources. This information is not currently handled with the Resource Model, and can not thus be taken into consideration during the matchmaking.

Despite these inaccuracies in the combined capability calculation and capability matchmaking, the presented approach can provide a valuable aid for the system designer. It gives a possibility to automatically explore large resource catalogues and rapidly filter out the unsuitable resources, leaving only the possible resources and resource combinations for the given requirement. The designer doesn’t need to do parameter or interface compatibility checking, but may concentrate on the layout planning and final resource selection. Consequently, less manual and time-consuming search and filtering effort is needed to find and evaluate different alternative solutions, and more alternative configurations can be considered, compared to traditional design approach, leading to potentially innovative system solutions. Thus, we expect the presented capability matchmaking approach and software to ease and speed up the system design and reconfiguration planning process.

In the future, new industrial projects will be established to test the matchmaking approach and software in wider industrial settings covering larger amount of different process capabilities. Furthermore, the reliability and information content of the matchmaking result should be increased by extending the current approach with automatic layout generation for feasibility checks and processing time estimations.

**Declaration of Competing Interest**

None

**Acknowledgments**

This research has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement no 680759 and project title ReCaM (Rapid Reconfiguration of Flexible Production Systems through Capability-based Adaptation, Autoconfiguration and Integrated Tools for Production Planning).
CRediT Author statement for paper

Eeva Järvenpää: Conceptualization, Methodology, Investigation, Writing, Supervision, Project Administration
Niko Siltala: Conceptualization, Methodology, Investigation, Software, Validation, Writing – Review and Editing
Otto Hylli: Methodology, Software, Validation
Minna Lanz: Supervision, Funding acquisition

References

Järvenpää, E., Siltala, N., Hylli, O., Lanz, M. 2018c. Utilizing SPIN rules to infer the parameters for combined capabilities of aggregated manufacturing resources. IFAC-PapersOnLine 51, 84–89.