Application of Capability-based Adaptation Methodology for a Small-size Production System

Eeva Järvenpää[1], Minna Lanz[2] and Reijo Tuokko[3]

Tampere University of Technology, Department of Mechanical Engineering and Industrial Systems, P.O. Box 589, FI-33101 Tampere, Finland

[1]eeva.jarvenpaa@tut.fi, [2]minna.lanz@tut.fi, [3]reijo.tuokko@tut.fi

Keywords: Adaptation, capability matching, capability model, production system, resource capability, resource description, resource ontology

Abstract. Today's production systems need to adapt rapidly to changing product requirements. Adaptation can be eased by formal models, representing product requirements and system capabilities, which convey the needed information for the adaptation planning activities. This paper presents an adaptation approach, which is based on matching the product requirements to the resource capabilities, and applies it to a practical case study in TUT-microfactory environment. The main emphasis is placed firstly on introducing the ontological resource description, which facilitates the representation of resource capabilities. Secondly, the rules, according to which the information about the product requirements and resource capabilities are compared to find the match, are highlighted. The proposed approach facilitate automatic filtering, reasoning and generation of alternative adaptation scenarios from a vast amount of resource information. As a result, less manual information handling and reasoning is required during adaptation planning.

1. Introduction

The operation environment of today's manufacturing companies is quickly evolving. Only thing that is evident is change. To survive in such a dynamic environment, the production systems need to be able to adapt rapidly to different requirements based on the demand. Modular, easily reconfigurable systems with standardized interfaces, have emerged to answer to this requirement (ElMaraghy, 2009; Ferreira et al., 2010). However, standardization of hardware and software interfaces is not, in itself, enough to support rapid adaptation of production systems in a constantly changing environment. Intelligent methods, tools and information models are also needed to help humans in the planning of this adaptation, or even allow reactive adaptation to take place while the system is running (Järvenpää et al., 2011).

Crucial factor for adaptation planning are formal models which convey the needed up-to-date product, process and system related information to the planning activities (Järvenpää, 2012). This paper presents an adaptation approach, which core lies on capability-based matching of product requirements and systems that are used to manufacture these. This, partly automatic reasoning, aiming to support human designer in

Copyright © 200x Inderscience Enterprises Ltd.

the planning activities, is enabled by the formalized representation of the resource capabilities and product requirements, as well as rules defining how the capability information should be matched against the product requirements. The paper first introduces the developed capability-based adaptation methodology and its related concepts. In the second part of the paper, this methodology is applied in a practical cell phone assembly case study implemented in the TUT-microfactory environment (TUT refers to Tampere University of Technology). Finally, section 4 concludes the paper.

2. Capability-based adaptation of production systems

This section will introduce a novel approach for adapting production systems by matching product requirements with the existing resource capabilities. First, the meaning of the term 'adaptation' will be discussed. In the second section the resource description based on capability modelling will be introduced, followed by the discussion on how the capabilities are matched with the requirements.

2.1. Definition of adaptation

Wiendahl and Heger (2004) identified five types of changeability of manufacturing systems: reconfigurability, changeoverability, flexibility, transformability and agility. Later on Wiendahl et al. (2007) used changeability as a general term as a characteristic of a system to accomplish early and foresighted adjustment of the factory's structures and processes on all levels to change impulses economically. Based on the literature around flexible, reconfigurable and adaptive manufacturing (ElMaraghy, 2009; ElMaraghy, 2006; Koren, 2010; Mehrabi et al,. 2000; Tolio & Valente, 2006), it is difficult to completely differentiate these concepts. Flexibility is often referred to 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, 2006). However, these definitions can be used only if the boundary of the system is clearly defined. Tolio and Valente (2006) stated that depending on the border, the type of changeability can be interpreted as reconfigurability or as flexibility and therefore, it is not possible to define general statements for these characteristics. ElMaraghy (2006) divided the manufacturing system reconfiguration into both physical and logical reconfiguration touching those both definitions of flexibility and reconfigurability.

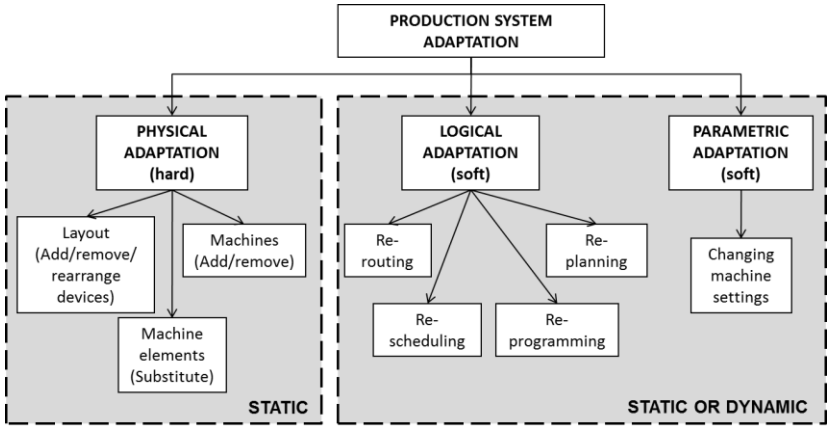

Figure 1. Types of production system adaptation, modified from (ElMaraghy, 2006).

In the context of this research, the term "adaptation" is used to refer to all controlled changes the production system goes through during its lifecycle. The adaptation can be either physical (e.g. changing the layout of the system, adding or removing machines or machine elements), logical (e.g. changing the process sequence, re-routing or re-scheduling production) or parametric (changing the adjustable machine parameters, e.g. speed of a conveyor line). Figure 1 represents and explains these three types of production system adaptation.

Adaptation can also be divided into static and dynamic adaptation. Static adaptation is the change of system design during downtime of the system. It needs planning, either by human or through automatic planning methods. Dynamic adaptation is the change of system design during the operation of the production system. These dynamic changes include either logical or parametric adaptation. Dynamic adaptation allows the production system to react to changes in its environment in real-time, for example to recover from disturbances on the line and to self-organize itself to balance the production flow. Whereas physical adaptation is usually done on the static level, the logical and parametric adaptation can be either dynamic or static. This means that logical and parametric changes can be executed while the system is running or during its downtime. The developed adaptation methodology aims to support both static and dynamic adaptation.

2.2. Resource description based on capabilities

Each device in the production environment has certain properties and behaviors. Some of these properties and behaviors allow the device to perform certain functions. The properties of the devices have certain ranges, and the functionality of the devices is restricted by certain constraints. These can be, for example, environmental constraints like the maximum permitted humidity and temperature of the operating environment, or technical properties, such as the maximum torque of a spindle or the velocity range of a moving axis. Automatic matching of available devices against product requirements requires formalized and structured representations of the functional properties and constraints of the devices.

Most of the traditional approaches to resource description tend to classify the resources into groups based on their common properties, or the functions they provide (e.g. milling machines, lathes, robots and so on). Unfortunately this kind of classification limits the expressiveness of the representation. To overcome this limitation, instead of classifying devices, in the proposed approach the functional capabilities of the devices are classified. This way, one device may have multiple capabilities which can be used in different contexts. Furthermore, new capabilities may be discovered during the resource lifecycle and these can be assigned to the device when, and if, they emerge. In the proposed approach the capabilities are functionalities of resources such as 'drilling', 'screwing', 'moving' and 'grasping'. For the capability classification, the capability taxonomy has been constructed so that it allows reasoning between different levels of abstraction.

The following sections will first shortly highlight the existing approaches, which use capability descriptions as a basis for matching product requirements against resource charasteristics. After that, in the subsequent sections, the developed capability modelling approach and resource ontology will be discussed in detail.

2.2.1. Existing approaches for capability description

Few other researchers have also applied capabilities to describe resources and allow rapid resource allocation, system reconfiguration or self-organisation. Timm et al. (2003; 2006) proposed an ontology-based capability management approach for multi-agent-based manufacturing systems. The capabilities were represented based on 1st order logic and the capability concepts were organized in taxonomic hierarchies. Lohse et al. (2006) presented an ontological equipment model, which follows function-behaviour-structure framework and includes separate formalism for the specification of a module's capability and interfacing requirements. Barata et al. (2008) and Cândido and Barata (2007) presented a multi-agent-based control architecture for a shop floor system (CoBaSa). They used ontology to model basic skills and complex skills, where complex skills are aggregations of the basic skills. Ameri and Dutta (2008) connected buyers and sellers of manufacturing services in a web-based e-commerce environment by formal ontology describing the manufacturing services in terms of capabilities. Smale and Ratchev (2009) proposed a capability-based approach for multiple assembly system reconfiguration. In general, the limitation of these approaches is that they lack either the ability to model combined capabilities of combinations of resources, or if this feature exist, they don't incorporate parameter information into the capability definition.

2.2.2. Introduction to the capability modelling

Figure 2 illustrates the capability related terms in the proposed capability modelling approach. Capabilities are composed of two components: capability concept name and capability parameters. The functionality of the resource, such as "screwing" is described by the capability concept name. The capability parameters represent the technical properties and constraints of resources, such as 'speed', 'torque', 'payload', and so on. For example, capability with concept name 'moving', has parameters 'speed' and 'acceleration'. In other words the concept name of the capability indicates the operational functionality of the resource, whereas the parameters of the capabilities distinguish between capabilities having the same concept name. The capability parameters allow

determining which resource has the capability that best fits to the given product or production requirement. Capabilities are divided into simple capabilities and combined capabilities, where the combined capabilities are combinations of simple capabilities, usually formed by combinations of devices. The numbers by the arrows in Figure 2 indicate that e.g. combined capability is composed of two or more simple capabilities.

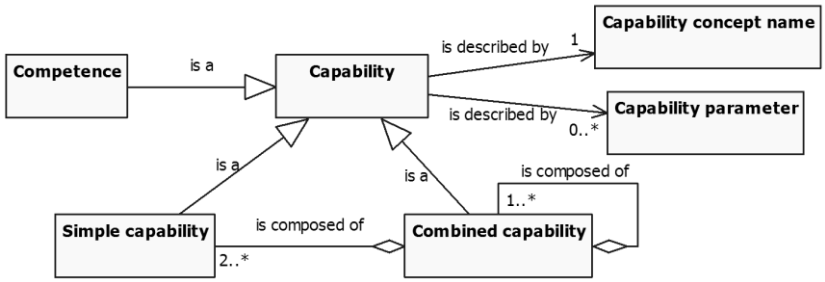

Figure 2. UML diagram of capability related terms in the proposed approach (Järvenpää, 2012).

2.2.3. Formal ontology for describing resources

In the proposed approach, ontological modelling is used to formalize the representation of the resources, capabilities, system configurations and product requirements. Ontologies play an important role in knowledge-based modelling. They provide a standardized, both machine and human interpretable, way to present knowledge from different domains and from heterogeneous knowledge sources. Ontologies are developed to support the exchange of meaningful information across autonomous entities that can organize and use the information heterogeneously. According to Gruber (1993), "an ontology is a formal, explicit specification of a shared conceptualization". The conceptualization is applied to a limited domain, such as the product, process and system domains. The conceptualization aims to break down the different terms and entities of this domain into well-defined and distinctive concepts. The concepts are expressed in a formal way in order to allow computers to use them. The concepts have to be explicit in order to avoid inconsistencies and ambiguities in meaning. This can be achieved by using non-ambiguous classifications, relations or metrics. Lastly, the conceptualization has to be shared and agreed upon by the different user groups in order to provide a common means of communication and frame of reference. (Gruber 1993.)

The Core Ontology, originally developed by Lanz (2010) for product-process-system representation, has been extended for describing the capabilities of the resources, resource interfaces, as well as lifecycle information relating to resources and certain processes. The ontology and its instances are saved into a Knowledge Base (KB), described in detail in (Lanz, 2010). The ontology has been built with the Protégé OWL-DL tool. OWL-DL is based on description logics and it allows the domain concepts to be largely defined according to a predefined formalism. (Stanford Center for Biomedical Informatics Research, 2012.)

Author

2.2.4. Capability model

The proposed capability modeling method relies on capability modularization. The approach is based on functional decomposition of upper level combined capabilities into simple capabilities and assigning these simple capabilities for individual resources in a modular way. When multiple resources are combined, the simple capabilities form combined capabilities. The systematic design approach of Pahl & Beitz (1996) provides a fundamental relationship between function and function decomposition. Function decomposition represents how a function is achieved through a set of sub-functions, which are finer-grained functions. (Pahl & Beitz 1996.) Similarly, the combined capabilities are achieved from finer-grained simple capabilities. Functional decomposition allows the definition of *is_part_of* relations between the simple and combined capabilities.

In the ontology, the combined capabilities are modelled using capability associations as links between the simple and combined capabilities. In the resource ontology, the devices are assigned the simple capabilities that they posses. Based on the defined capability associations, the device combinations contributing to certain combined capability can be identified and queried. Of course, the devices also need to have matching interfaces to be able to co-operate. Figure 3a shows how the capability associations are used to form combined capabilities from the simple ones. Figure 3b illustrates an example of capability associations by 'transporting' capability in case of a robot unit consisting of a robot and gripper. In order to have "transporting" capability, both moving and holding associations need to be satisfied. The robot alone has only the ability to move its joints within a workspace. When combined with a suitable gripper (having 'grasping' capability), together they are able to transport pieces from one place to another. Also conveyor alone would have the 'transporting' capability, because it can move within a certain workspace and hold items either by a specific jig or by gravity on the belt. More examples of simple, combined and resource specific capabilities in the TUT-microfactory environment, can be seen in Section 3.

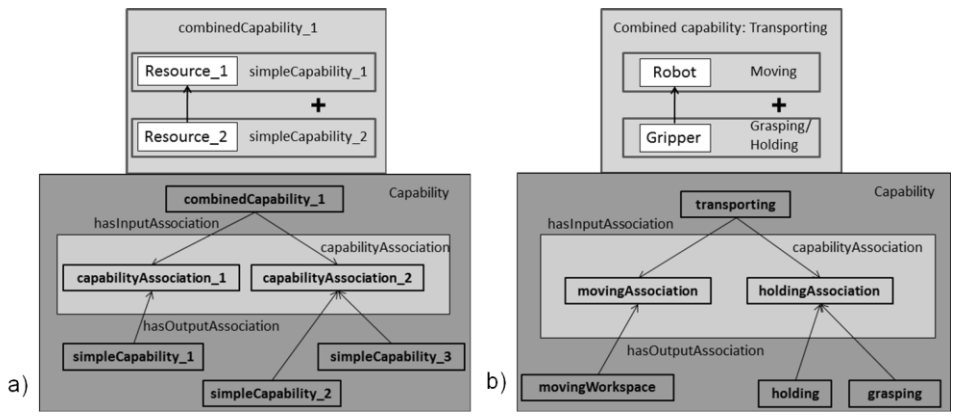

Figure 3. a) Model for combined capabilities; b) Sample of the instantiated capability model (Järvenpää 2012).

Generally, in modularization, the interactions between the involved components are to be minimized (Lehtonen, 2007). In the case of capabilities, the functional decomposition aims to minimise the redundancy and interdependencies between the parameters of the simple capabilities, so that the parameters fit naturally under one capability concept name. Also the assignment of the same parameters for multiple different capabilities is minimized. This fit is based on production engineering domain knowledge. It is recognized that this "natural fit" is not realistic in all the cases. Therefore, some artificial simplifications are made during the creation of the instantiated capability model. For more information about capability parameters, please refer to (Järvenpää 2012). Secondly, the capability modularization aims to provide reusability of the capabilities among different types of resources.

Capability model defines the generic capabilities, i.e. a pool of capabilities that can exist in a system. When these generic capabilities are assigned to the resources, they become resource specific when filled with resource specific parameter values. The capabilities are linked to the capability taxonomy, which organizes the capabilities in different abstraction levels, e.g. 'milling' is a sub-class of 'materialRemoving' capability in the taxonomic hierarchy (Järvenpää, 2012).

2.2.5. Device blue prints and Individual devices

As known, the production environment is constantly changing, and the condition and capabilities of the resources evolve during their individual lifecycles and usages. In adaptation context the resources have already been operating, which often indicates that the capability of the resource is not anymore the same as it was when the resource was taken to use. Therefore, the description of the resource has to be updated over time. For this reason, devices have two separate, but linked representations within the resource ontology: device blue prints and individual devices. Figure 4 shows the relations between the device blue prints and the individual devices, and their associated information elements.

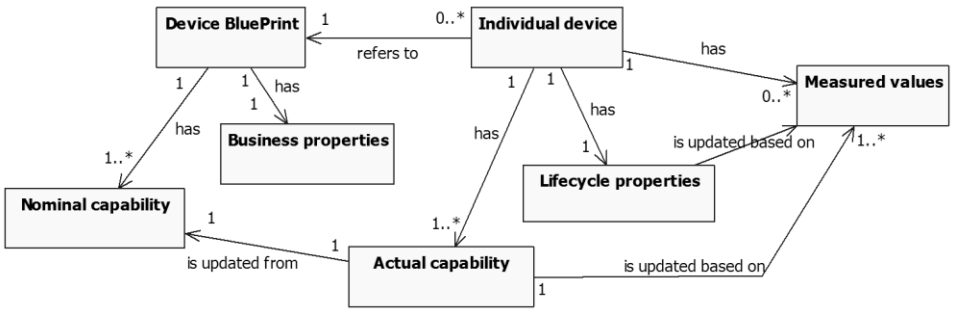

Figure 4. UML diagram of device blue prints and individual devices (Järvenpää et al. 2012).

The device blue print describes the capabilities and properties of one type of device, as given in the suppliers' catalogues. This is the nominal capability of the device. The individual devices are presented in a separate class which refers to the blue print device, yet presents the actual capabilities of the particular, individual resource which exists on the factory floor. The numbers by the arrows in Figure 4 indicate, that e.g. one device

blue print can have zero or multiple individual devices referring to it. The individual devices have actual capabilities, which are affected by the lifecycle of each individual device and updated according to measured values from the factory floor. For example, if the measured accuracy of the machine differs from the value defined in the nominal capability, this updated value can be given in the actual capability definition. Maintenance and service operations or adaptations done to the resource can also change its capability. (Järvenpää et al. 2012.) In capability-matching the up-to-date (actual) capability information should always be used if available.

2.3. Capability-based matching framework

In the proposed capability-based adaptation approach, the product requirements are matched against the system capabilities and the system is adapted, until it satisfies the requirements of the product, i.e. until it is compatible with the new product requirements (Järvenpää, 2012). The product requirements can be extracted from a 3D product model with a feature recognition and pre-process planning software, as has been presented in (Garcia et al. 2011), or they can be formulated manually and then saved to the Knowledge Base as will be exemplified in this paper. The pre-process plan is an ordered graph of generic activities referring to specific levels on the capability taxonomy (Garcia et al., 2011). The activities in the pre-process plan can refer either to more abstract level capability, such as "materialRemoving", or to more detailed level capability, like "drilling". The capability matching is performed according to the capability taxonomy and capability matching rules which connect the product and resource domains to each other. Figure 5 represents the framework of the capability-based matching.

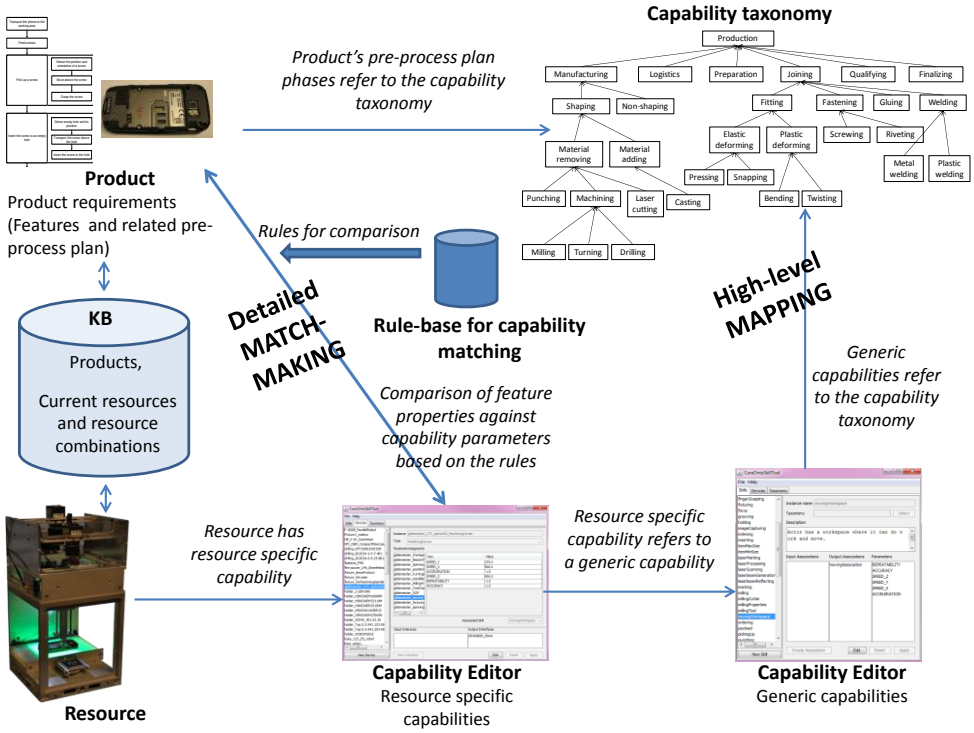

Figure 5. Framework for matching the product requirements against the system capabilitied, modified from (Järvenpää, 2012).

The taxonomy, included in the Core Ontology, is used to make a crude search that maps the resources with the required capabilities at a high level (capability concept name level), whereas the detailed reasoning with the capability parameters and combined capabilities is based on rule-based reasoning. Rule-base is developed as a store for the rules used in the capability matching. An extensive Python framework was developed for writing the rules and for retrieving the information relating to the product requirements and resource capabilities from the Knowledge Base. Three types of rules have been defined: 1) Domain expert rules - Rules for defining how the capability and its parameter information are applied in different domains when comparing with the product requirements; 2) Combined capability rules - Rules for reasoning out the parameters of the combined capabilities and; 3) Adaptation rules - Rules indicating how other criteria, such as availability and scheduling, device condition and lifecycle, as well as user and company specific criteria is used in the final resource selection and configuration generation. (Järvenpää, 2012.) In the following section this adaptation methodology is applied to a TUT-microfactory environment.

3. Practical case study – TUT-microfactory environment

The case study aims to describe how the capability-based adaptation methodology is applied and especially how the capability matching rules are used in practice, and provide

a proof of concept of the developed methodology. In this context the microfactory system can be viewed as a static system where the adaptation takes place "offline", based on human-centric planning. In the microfactory environment, modularization and standardized interfaces are the enablers for the adaptation. The main task in the following case study is to evaluate if the existing TUT-microfactory system has the capabilities to cope with the requirements set by the selected case product assembly. First, an introduction to the TUT-microfactory concept will be given. Secondly, the product requirements are defined, followed by the definition of existing microfactory system capabilities. After that, the matching of the product requirements against the system capabilities based on the capability matching rules will be explained in detail.

3.1. Introduction to TUT-microfactory concept

The TUT-Microfactory is a modular construction kit type concept with easy and rapid reconfigurability for different manufacturing processes of hand held size, or smaller, products. The system structure is designed with an idea that a base module can work as an independent unit including all the needed auxiliary systems. The outer dimensions of one base module are 300 x 200 x 220 mm and the inside workspace is 180 x 180 x 180 mm. The base module includes a clean room class work space, a control cabinet and the equipment needed by the clean room. Since the production module does not need a separate control cabinet, the factory can be aggregated fast and easily on a desktop table or other flat surface. This and small size of the modules enable extreme mobility of the production capacity. (Heikkilä et al., 2010.)

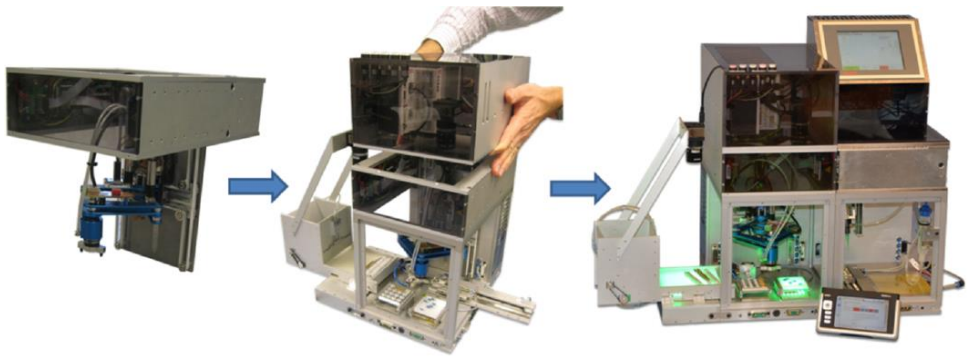

Figure 6. Plug-and-play interfaces for easy configuration of a complete TUT-microfactory system.

The production module can be tailored to certain processes by placing process modules on top of the base module (Figure 6). Process module can be e.g. a robot, laser or machining unit. In addition to the top side of the base module, both sides and the front side can be left open when adjacent cells compose one integrated work space. Feeders and other devices can be placed in the opening on the sides. (Heikkilä et al. 2010.)

3.2. Definition of the product requirements

The case product is a cell phone body, in which four screws are to be attached (Figure 7a). The product requirements are shown in the graph in Figure 7b. On the left-hand side

is the process plan and on the right-hand side are the product characteristics, which affect the required capabilities in each process phase. Additionally, the requirements, which are more project and user-preference related rather than product related, such as the required speed for transporting the phone and the desired type of feed, are shown in the figure. The steps in the process plan have a direct link to the capability taxonomy, which allows the mapping between the product requirements and existing system capabilities at the concept name level.

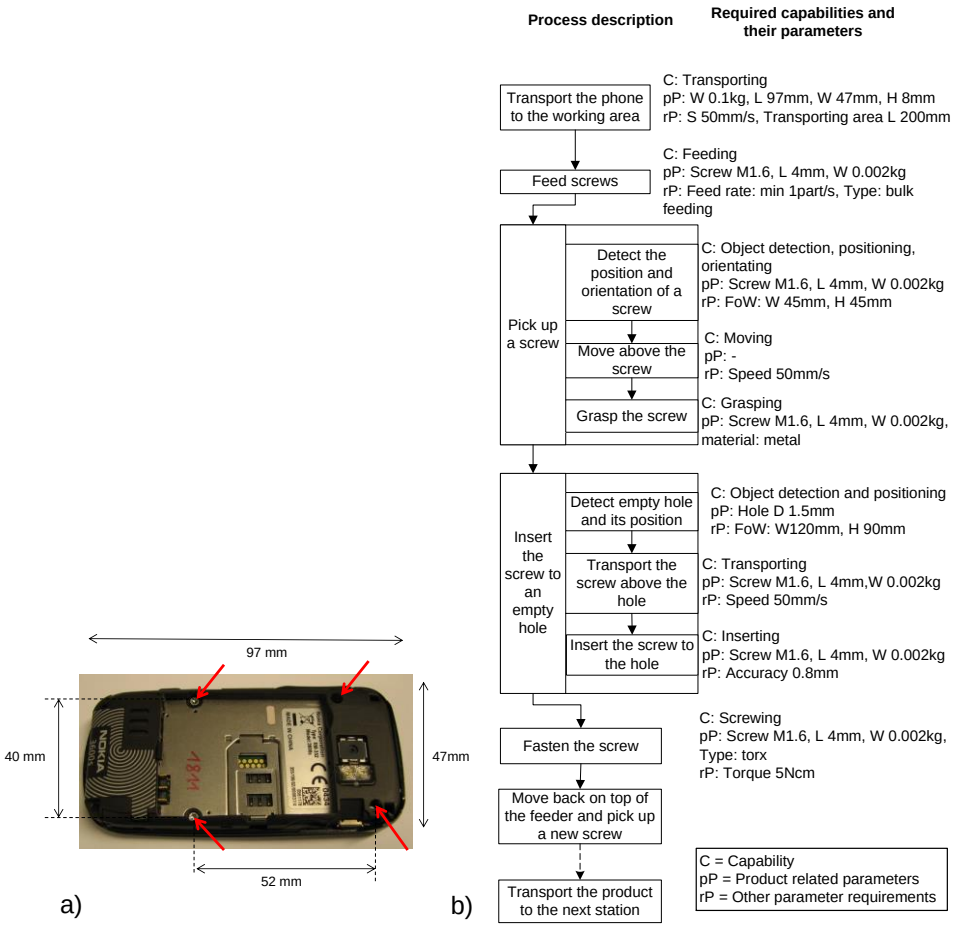

Figure 7. Case product; b) Pre-defined process plan and related capability requirements, modified from (Järvenpää et al., 2013).

The system architecture of the TUT-microfactory concept places some constraints on the possible layout and configuration of the modules. In the TUT-microfactory concept, the microfactory frame is needed to contain the process module and auxiliary devices. For the case study, one pre-condition is that the product should be produced with the TUT-microfactory. Therefore, the constraints set by the system architecture need to be considered in the capability matching. For example, as the width of the TUT-microfactory module is known, then the transporting distance of the cell phone from one side of the module to the other is pre-defined. Some of these requirements are set by the

selection of other devices in the system. For instance, the required field of view (FoV) of the camera units are not only determined by the product size, but also by the means of transportation and feed for the product and parts. In other words, the selection of some devices propagates some further new requirements. Examples of these kinds of requirements are included in the case example and are also shown in the Figure 7.

3.3. Definition of the system capabilities

The existing system consists of a TUT-microfactory module, a cartesian manipulator, a screwdriver unit, a feeding system, a belt conveyor and a machine vision system with 2 camera units (see Figure 8). Figure 9 shows the capabilities of the existing microfactory system. The devices are grouped in their natural combinations, e.g. the camera unit consists of camera and optics, whereas the machine vision system consists of the camera unit, a PC and ambient lighting. The same grouping is also used for the description of the system in the resource ontology. The arrows in Figure 9 indicate to which combined capability the simple capabilities of the devices contribute.

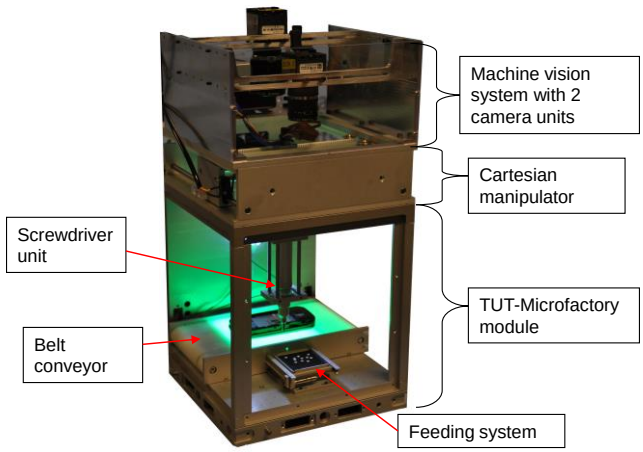

Figure 8. Existing TUT-microfactory system.

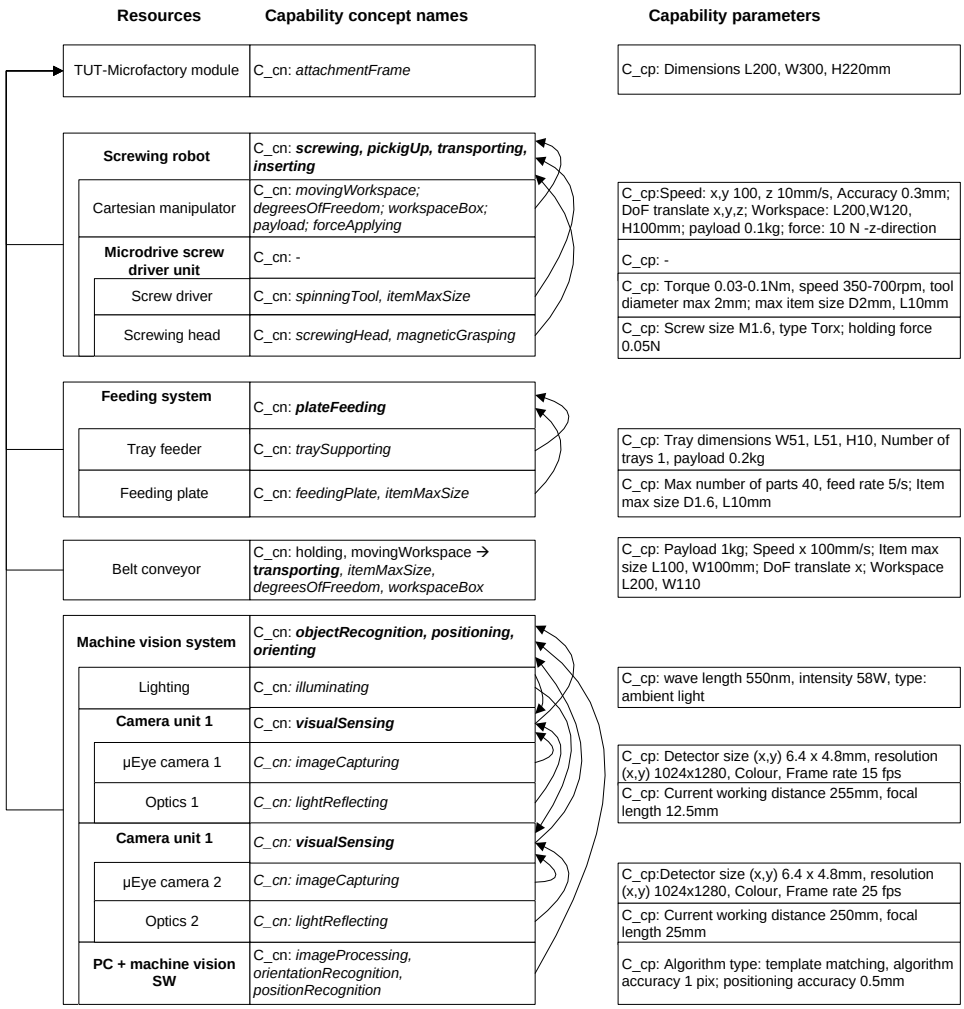

Figure 9. Capabilities of the existing TUT-microfactory system for flexible screwing.

In order to save space and ensure readability, only those parameters needed in the reasoning of this particular matching case, are shown in the Figure 9. To illustrate more intuitively the capability parameters and the formation of the combined capabilities from the simple capabilities, Figure 10 shows the capabilities of the screwing robot consisting of Cartesian manipulator and screw driver unit, which again consists of a screw driver and a screwing head.

Author

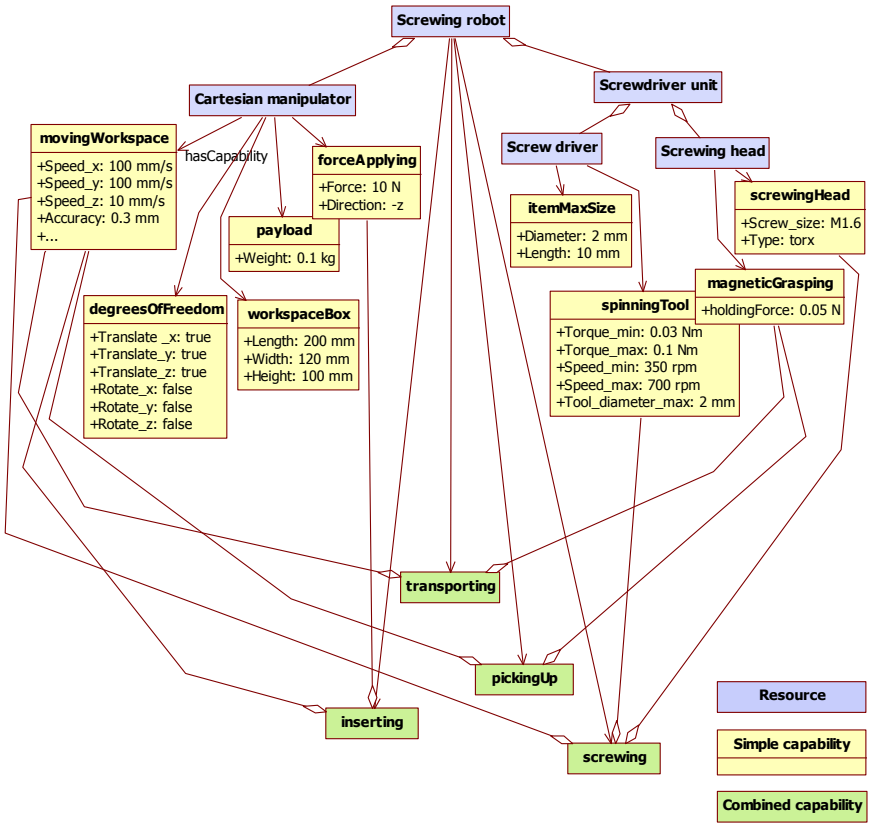

Figure 10. Capabilities of a screwing robot included into the TUT-Microfactory screwing cell, modified from (Järvenpää et al., 2013).

3.4. Matching the product requirements against the system capabilities

Based on the description of the product requirements and existing system capabilities, the capability taxonomy, and the rules for the detailed capability matching, it is possible to reason out if the existing system has all the required capabilities needed to perform the screwing operations. The steps in the process plan have a direct link to the capability taxonomy, which allows the mapping between the product requirements and existing system capabilities at the capability concept name level. This high-level mapping results, that all the required capabilities, at the concept name level, can be found from the existing TUT-microfactory system. Next, the detailed capability matching needs to be performed based on the rules in the rule-base. In the following paragraphs, these resoning procedures are explained step-by-step.

Step 1: Transport the phone to the working area

The high-level capability mapping detects two devices (combinations) from the system which have the "transporting" capability. These are the belt conveyor and the screwing robot. The detailed matching first checks whether the product size and weight are suitable

for the current devices (RULE 1a), and secondly it checks if the workspace of the device and the speed of the transporting capability matches the requirement (RULE 1b). The rules used for the matching are shown in Figure 11.

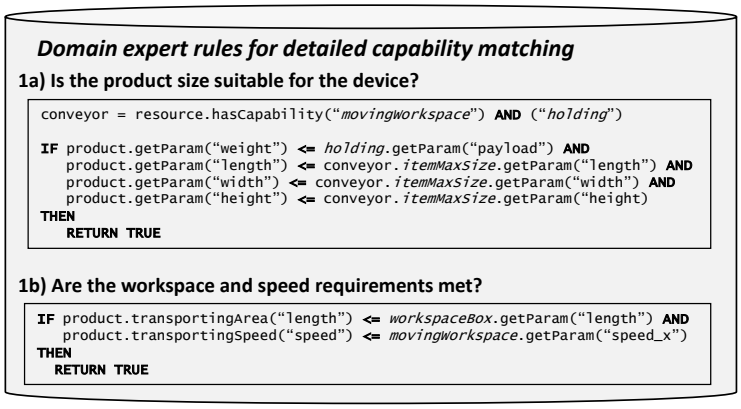

Figure 11. Rules for step 1.

When these rules are applied to the presented case, i.e. filled with the parameter values shown in Figure 9 and Figure 8, it reveals that the belt conveyor capabilities match the requirements. Only the rules for the conveyor case are shown here. However, similar reasoning with the screwing robot is shown later in Figure 13. Those rules would immediately reveal that the screwing robot is not a suitable device combination for transporting the cell phone, because of the size, weight and material of the product.

Step 2: Feed screws

Based on the high-level capability mapping, one device combination in the current system has the capability "plateFeeding". This is the feeding system. The designer has specified that the screws should be fed by a bulk feeding method in order to ease the manual handling of the screws. According to the capability taxonomy, the "plateFeeding" is a specialization of "bulkFeeding" and therefore fulfils the requirement. Because bulk feeding is a method which doesn't provide the parts in a certain position and orientation, the machine vision system, or another system providing "objectRecognition", "positioning" and "orienting" capabilities is required to be able to detect the parts that can be picked up from the feeder. These capabilities are also available in the system. The rules for detailed capability matching are used to find out if the existing feeding system is able to feed the screws. First, it needs to be checked if the part size is suitable for the feeder (RULE 2a).

The matching shows that the current feeding system is able to feed the screws. However, the position and orientation of the screws also need to be detected in order to feed the parts in a specified position and orientation. The high-level mapping finds that the machine vision system consisting of two cameras has the capabilities of "object detection", "positioning" and "orienting". The physical arrangement of the machines, acquired from the virtual model, defines the position of the camera units in relation to the feeder and conveyor. Camera unit 1 is above the conveyor and Camera unit 2 is above the

feeder. The working distances of the cameras are pre-defined based on the current installation.

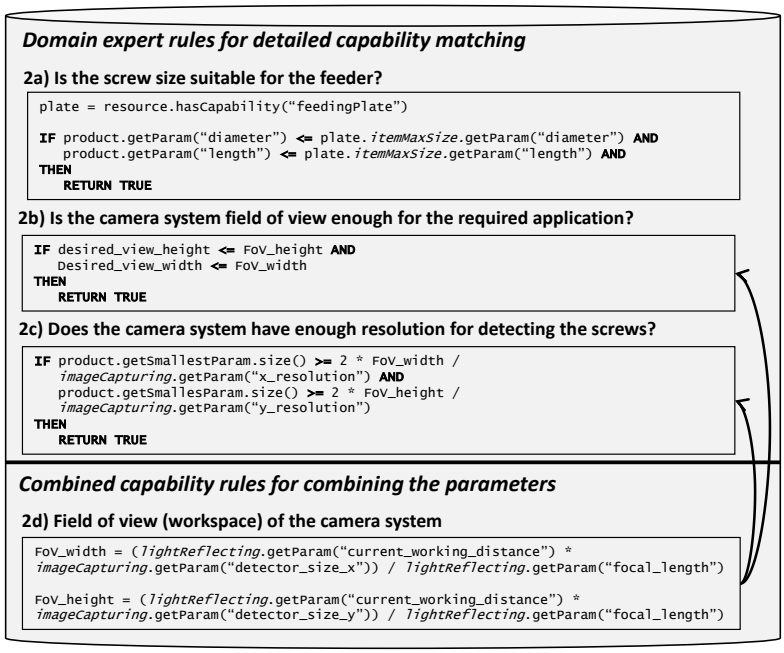

Figure 12. Rules for the step 2.

The field of view (FoV) of the camera system is calculated based on the working distance of the camera, the detector size (CCD width x CCD length) and the focal length of the optics as defined by the combined capability rule (RULE 2d). This results that FoV of camera unit 1 is 130 x 98 mm and FoV of camera unit 2 is 60 x 48 mm. Rules 2b and 2c are then used to determine if the camera system field of view is large enough for the application and that its resolution is enough to detect the screws from the feeder. The desired field of view is defined after the feeding plate has been selected. Based on the plate size, the desired FoV is 45 x 45 mm. Camera unit 2 does have a bigger FoV, so it is suitable. The minimum required detector resolution is calculated with the Nyquist principle based on the field of view and the smallest detectable feature, as defined by RULE 2c. As the screws to be detected are 1.6 mm, the minimum detector resolution is 75 x 60 pixels. The detailed capability matching shows that the camera resolution goes well beyond the required resolution.

Step 3: Pick up, insert and fasten the screw

The high-level mapping finds one device combination having the "pickingUp", "inserting" and "screwing" capabilities. This is the screwing robot consisting of the cartesian manipulator and the screwdriver unit, as shown in Figure 10. The detailed capability matching needs to check if the screws can be picked up with the current device combination (RULE 3a), whether the transportation capability fullfils the set speed requirements (RULE 3b), and whether the screw driver is able to fasten the specific screws used in this case study (RULE 3c). RULE 3a uses the information provided by the RULE 3d, which calculates the payload of the robot + screw driver combination.

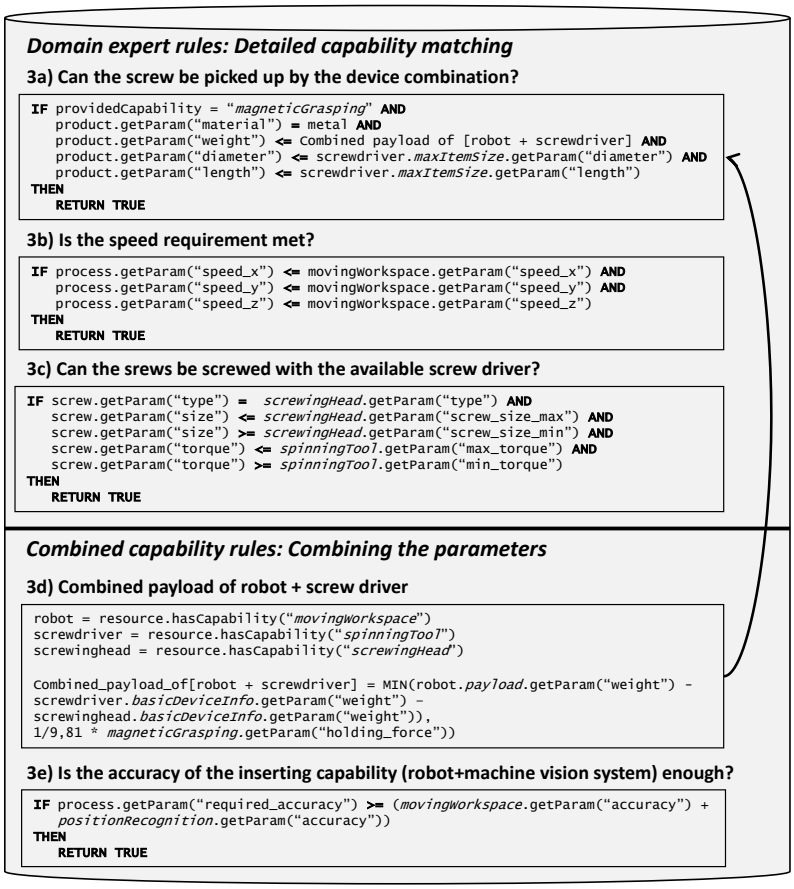

Figure 13. Rules for the step 3, modified from (Järvenpää et al. 2013).

Based on the detailed capability matching, the robot-screwdriver combination is able to pick up the screw and move at the desired speed. Because the transportation of the cell phone on the belt conveyor doesn't position and orient the product in the system, the positions of the holes are not known. Therefore, a method to detect the empty hole and its position has to be available in the screw insertion phase. As discussed earlier in the step 2, the machine vision system has these capabilities. The detailed-level matching considering the field of view and resolution requirements is carried out as was done earlier. Camera unit 1 is able to fulfil the given requirements.The RULE 3e shows that the combined accuracy of the machine vision system and the screwdriver robot is, in the worst case scenario, 0.8 mm. This was the original requirement, which means that the accuracy requirements are fulfilled.

Finally the detailed matching checks whether the screw type and size, as well as the required torque, are suitable for the existing screwdriver (RULE 3c). The matching shows that the screws can be fastened with the existing screw driver. However, the screw driver is only suitable for one size of screws. If the screw size changes, it will immediately require physical adaptation to the system.

Step 4: Adaptation scenario – changing the screw size
In this case scenario, the screw size changes from M1.6 to M2. When changing the screw size, the required capabilities remain the same at the concept name level. Only the screw size parameter is changed. The current screwdriver is suitable for screwing only size M1.6 screws, which means that, based on RULE 4a, a new screwing head needs to be attached to the screwdriver (Figure 14). The rule 4b is used to define whether the new screwing head is compatible with the given screwdriver.

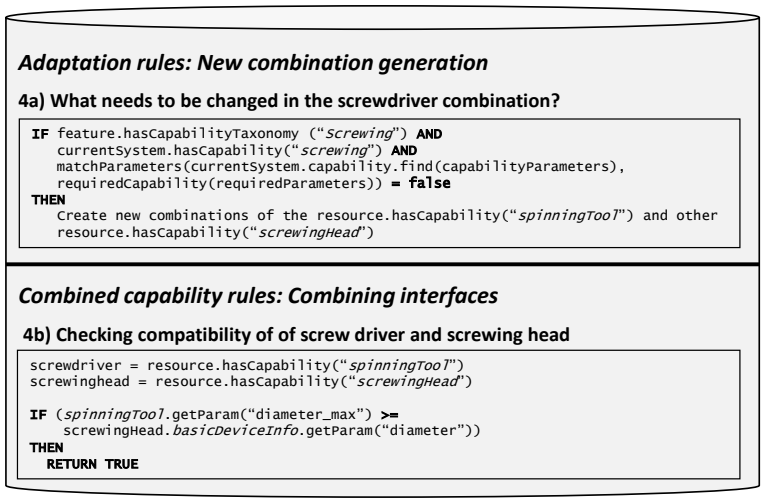

Figure 14. Rules for the step 4, modified from (Järvenpää et al., 2013).

If the screwdriver head is not compatible with the screwdriver, for example in this case if the screwing head is for bigger screws than M2, the whole screwdriver needs to be changed. Small changes in the product design can, in this case, be handled with relatively small changes to the system. However, this example illustrates well that when certain border constraints of the capabilities are crossed, the magnitude of the adaptation can grow significantly due to the propagation of the changes. In this example case, as long as the required screw size is between M1 and M2, the system can be adapted by just changing the screwdriver head. If bigger screws are used, the whole screwdriver needs to be replaced. And again, these changes may be propagated to the robot if, for example, the new screwdriver is not compatible with the robot interface.

4. Results and Conclusions

This paper presented capability-based adaptation methodology, which aims to support rapid adaptation of production systems both in static and dynamic adaptation contexts. The proposed adaptation methodology allows automatic generation of system configuration scenarios based on given product requirements and supports rapid allocation of resources and adaptation of systems. From the information management point of view, it supports automatic filtering of information and finding suitable solution proposals from a large search space, thus reducing the manual information-processing

required during adaptation planning. Furthermore, the resource model provides up-to-date information about the resources, facilitating more reliable adaptation plans. In a small factory with only a few resources, management of the resource information is not a problem. In large factories and production networks, automatic management and filtering of this information has substantial potential for reducing the amount of manual work, and thus reducing the time used for planning activities.

Compared to the existing capability-based adaptation approaches, the presented methodology provides finer granularity to the resource description by separating the simple and combined capabilities, and thus allows the adaptation on a single tool level, not only on a machine level. In comparison to those methods, which do separate the simple and combined capabilities, this methodology takes one step further by incorporating the parameter information to the capability description. It operates with more details and therefore facilitates more detailed decision making relating to the system design and adaptation.

The case study presented in this paper showed how the developed capability-model, rule-base and the overall adaptation methodology can be used in practice to support human-controlled adaptation planning. The case study proved that the developed ontological resource description can be used to describe the capabilities of real production systems and the developed rules can be applied to make the match between the resource and product descriptions. The capability matching actions, discussed in the examples, can be done automatically based on these rules. However, during the study, the initial assumption, that humans cannot be removed from the adaptation planning process, was confirmed. For example, the proposed approach is not able to automatically handle the propagation of requirements, when the selection of one resource creates further requirements for the other resources, but these need to be controlled by the human designer. Secondly, when operating in a changing and complex environment, where the quantity and quality of the input information varies case by case, human intelligence should not be replaced by computers. Instead, both should be utilized in appropriate situations. Computers are good at handling, processing and filtering large amounts of data, whereas humans are capable of making intelligent decisions, which often require tacit, experience-based knowledge not managed by computers. Therefore, this paper proposed a computer-aided, capability-based adaptation methodology in which the computer's processing power is used for filtering the resource information from a large search space based on the given product requirements. The human expert can use his/her intelligence to control the reasoning process, check the feasibility of the proposed scenarios and to select the best one based on the specified criteria.

References

[1] Ameri, F. & Dutta, D., 2008. *A Matchmaking Methodology for Supply Chain Deployment in Distributed Manufacturing Environments*. Journal of Computing and Information Science in Engineering, 8(1), p. 9, ISSN 1530-9827.

[2] Barata, J., 2006. *The Cobasa Architecture as an Answer to Shop Floor Agility*. In: Kordic, V., Lazinica, A. & Merdan, M. (eds.), Manufacturing the Future. InTech, pp. 31-76, ISBN 3-86611-198-3.

[3] Cândido, G. & Barata, J., 2007. *A multiagent control system for shop floor assembly*. In: Marik, A., Vyatkin, V. & Colombo, V. (eds.), Holonic and Multi-Agent Systems for Manufacturing. Heidelberg: Springer, pp. 293–302, ISBN 3-540-74478-9.

[4] ElMaraghy, H. A., 2006. *Flexible and reconfigurable manufacturing systems paradigms*. International Journal of Flexible Manufacturing Systems, 17(4), pp. 261-276, ISSN 0920-6299.

[5] ElMaraghy, H.A. (ed.), 2009. *Changeable and Reconfigurable Manufacturing Systems*. London: Springer London, p. 405, ISBN 978-1-84882-066-1

[6] Ferreira, P., Lohse, N. & Ratchev, S., 2010. *Multi-agent Architecture for Reconfiguration of Precision Modular Assembly Systems*. In: Ratchev, S. (ed.), Precision Assembly Technologies and Systems. Springer, pp. 247–254, ISBN 3-642-11597-7.

[7] Garcia, F., Lanz, M, Järvenpää, E. & Tuokko, R. 2011. *Process planning based on feature recognition method*. In IEEE International Symposium on Assembly and Manufacturing (ISAM), 2011, p. 5.

[8] Gruber, T., 1993. *A translation approach to portable ontology specifications*. Knowledge acquisition, 5(2), pp. 199-220.

[9] Heikkilä, R., Järvenpää, E. & Tuokko, R. 2010. *Advances in the TUT Microfactory Concept Development*. Int. J. of Automation Technology, Vol. 4, No. 2, 2010, pp. 117-126.

[10] Järvenpää, E., Luostarinen, P., Lanz, M. & Tuokko, R. 2011. *Presenting capabilities of resources and resource combinations to support production system adaptation*. In IEEE International Symposium on Assembly and Manufacturing (ISAM), 2011, p. 6.

[11] Järvenpää, E., Luostarinen, P., Lanz, M. & Tuokko, R., 2012. *Adaptation of Manufacturing Systems in Dynamic Environment Based on Capability Description Method*. In F. Abdul Aziz (ed.), Manufacturing System. InTech, p. 26, ISBN 978-953-51-0530-5.

[12] Järvenpää, E. 2012. *Capability-based Adaptation of Production Systems in a Changing Environment*. PhD Thesis, Tampere University of Technology. 190 p. ISBN 978-952-15-2928-3.

[13] Järvenpää, E., Lanz, M. & Tuokko, R. 2013. *Capability-based Adaptation of Production Systems – Practical Case Study in TUT-microfactory Environment*. Key Engineering Materials, Vol. 572, 2014, pp. 245-248.

[14] Koren, Y. & Shpitalni, M., 2010. *Design of reconfigurable manufacturing systems*. Journal of Manufacturing Systems, 29(4), pp. 130-141, ISBN 0278-6125.

[15] Lanz, M., Rodriguez, R., Luostarinen, P. & Tuokko, R. 2010. *Neutral interface for assembly and manufacturing related knowledge exchange in heterogeneous design*

environment. In Precision Assembly Technologies and Systems, S. Ratchev (Ed.), Springer, pp. 21–29.

[16] Lanz, M., 2010. *Logical and Semantic Foundations of Knowledge Representation for Assembly and Manufacturing Processes*. PhD thesis. Tampere University of Technology, p. 138, ISBN 9789521523939.

[17] Lehtonen, T., 2007. *Designing Modular Product Architecture in the New Product Development*. PhD thesis. Tampere University of Technology, p. 220, ISBN 9789521518980.

[18] Lohse, N., Hirani, H. & Ratchev, S., 2006. *Equipment ontology for modular reconfigurable assembly systems*. International Journal of Flexible Manufacturing Systems, 17(4), pp. 301-314, ISSN 0920-6299.

[19] Mehrabi, M.G., Ulsoy, A.G. & Koren, Y., 2000. *Reconfigurable manufacturing systems: key to future manufacturing*. Journal of Intelligent Manufacturing, 11(4), pp. 403–419, ISSN 0956-5515.

[20] Pahl, G. & Beitz, W., 1996. *Engineering Design - A Systematic Approach*. 2nd edition., Springer-Verlag, Berlin/Heidelberg, p. 544, ISBN 3540199179.

[21] Smale, D. & Ratchev, 2009. *A Capability Model and Taxonomy for Multiple Assembly System Reconfigurations*. In B. Natalia (ed.), In 13th IFAC Symposium on Information Control Problems in Manufacturing. Moscow, pp. 1923-1928.

[22] Stanford Center for Biomedical Informatics Research, 2012. *Protégé ontology editor*. Available at: http://protege.stanford.edu/ [Accessed March 1, 2012].

[23] Timm, I. & Woelk, P.-O., 2003. *Ontology-based capability management for distributed problem solving in the manufacturing domain*. In: Schillo, M., Klusch, M., Müller, J. & Tianfield, H. (eds.), Multiagent System Technologies. Springer Berlin / Heidelberg, pp. 168-179, ISBN 978-3-540-20124-3.

[24] Timm, I.J., Scholz, T. & Herzog, O., 2006. *Capability-based emerging organization of autonomous agents for flexible production control*. Advanced Engineering Informatics, 20(3), pp. 247-259, ISSN 1474-0346.

[25] Tolio, T. & Valente, A., 2006. *An Approach to Design the Flexibility Degree in Flexible Manufacturing Systems*. In 16th Int. Conf. on Flexible Automation and Intelligent Manufacturing, FAIM 2006, June 26 – 28. Limerick, pp. 1229-1236.

[26] Wiendahl, H.P. & Heger, C.L., 2004. *Justifying Changeability – A Methodical Approach to Achieving Cost Effectiveness*. Journal for Manufacturing Science and Production, 6(1-2), pp. 33-40, ISSN 0793-6648.

[27] Wiendahl, H., ElMaraghy, H., Nyhuis, P., Zah, M., Duffie, N. & Brieke, M., 2007. *Changeable Manufacturing - Classification, Design and Operation*. CIRP Annals - Manufacturing Technology, 56(2), pp. 783-809, ISSN 0007-8506.