

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Adaptation of Manufacturing Systems in Dynamic Environment Based on Capability Description Method

Eeva Järvenpää, Pasi Luostarinen, Minna Lanz and Reijo Tuokko
*Tampere University of Technology, Department of Production Engineering
Finland*

1. Introduction

Nowadays manufacturing systems are characterized by constantly changing requirements caused by short lifecycle times of products, small batch sizes, increasing number of product variants and fast emergence of new technical solutions. Today's turbulent production environment calls for adaptive and rapidly responding production systems that can adjust to required changes both in production capacity and processing functions. The European level strategic goal towards Competitive Sustainable Manufacturing (CSM) asks for the re-use and adaptation of production systems (Jovane et al., 2009). Adaptation allows users to utilize the full lifetime and potential of the systems and equipment and in this way supports the sustainability, both from economic and ecologic perspectives. However, in the previous projects (e.g. (Harms et al., 2008)), it has been recognized that, because of often expensive and inefficient adaptation process, companies rarely decide to adapt their production systems. This is mainly due lacking or insufficient information and documentation about the capabilities of the current system and its lifecycle, as well as lack of extensive methods to plan the adaptation. Today the adaptation of production systems is practically a human driven process, which relies strongly on the expertise of the system integrators or the end user of the system.

Different manufacturing paradigms have been initiated in recent years to overcome the challenges relating to adaptivity requirements. Reconfigurable manufacturing systems (RMS) aim to meet these requirements by offering rapid adjustment of production capacity and functionality, in response to new circumstances, by rearrangement or change in their structure as well as in hardware and software components (ElMaraghy, 2006; ElMaraghy, 2009; Koren et al., 1999; Mehrabi et al., 2000). Agent-based and holonic systems take more dynamic approach to cope with the changeability requirements. Distributed Manufacturing System (DiMS) concept developed by Nylund et al. (2008) and Salminen et al. (2009) is based on holonic architecture, where the holons are autonomous entities able to communicate with other holons and form set of holons, holarchies, through common, well-defined, interfaces and negotiation process. In DiMS the production environment is seen as dynamic and evolving open complex system, where the decision making is based on negotiation process between these entities. Holonic manufacturing systems aim to offer a solution for changeability requirements by providing self-organizing capabilities. Whereas the reconfigurable system research focuses mainly on physical adaptation, in the latter approach

the adaptation is performed also on logical and parametric levels. However, no matter if the adaptation is happening on physical, logical or parametric level, intelligent methods and tools are needed to support efficient planning of adaptation.

A critical factor for computer-aided production system design and adaptation planning is efficient resource models, which provide the needed information for equipment selection and system integration. This chapter will introduce a novel method to formally describe the capabilities of resources to support the manufacturing system adaptation. First, in section 2, the term adaptation is defined. Section 3 reviews some existing approaches for describing resource capabilities. Section 4 will then describe the developed capability description method and in section 5 the mapping between product requirements and system capabilities will be covered. In section 6 case application of the capability descriptions in a holonic manufacturing framework, build into TUT heavy machining laboratory, will be discussed. Also the modular software system architecture of the holonic system will be introduced. Finally, the future work is discussed in section 7, followed by the conclusions in section 8.

2. Adaptation of manufacturing systems

This section aims to clarify the concept of adaptation in the context of this research. First, the traditional classification of different adaptation types is introduced. After that the Systems approach to adaptation is discussed. Finally, the relation between capabilities and adaptation is highlighted.

2.1 Classification of adaptation types

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, 2006; 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.

In order to avoid misunderstandings caused by the old definitions of reconfigurability, flexibility and other related terms, the term 'adaptivity' is used in this work, including both physical adaptation (reconfiguration) and logical adaptation. Besides those two types, also parametric adaptation is included into the definition. Fig. 1 represents and explains these three types of production system adaptation.

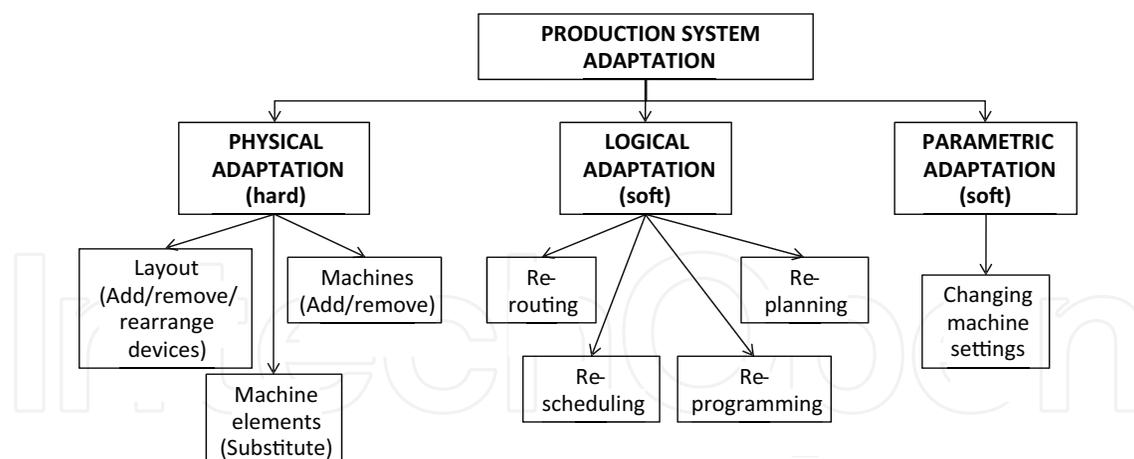


Fig. 1. Types of production system adaptation, modified from (ElMaraghy, 2006).

Adaptation can also be divided into static and dynamic adaptation. Static adaptation is the change of system design during downtime of the system. Dynamic adaptation is the change of system design during the operation of the production system. These dynamic changes be 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.

2.2 Systems approach for adaptation

Based on the systems approach, in order to achieve adaptability, the system must be able to learn from the experience (Bourgine & Johnson, 2006). The learning is achieved via gaining and understanding the feedback of change – its magnitude and direction. In order to understand the change, the system must be able to compare the past status with the new status of operations. Unfortunately, in the traditional operation environments the knowledge of neither the past nor the present status is in a computer interpretable and comparable form. It would require that the content, context and interaction between those is known. Without this content, context and their interaction information the adaptation to the changes in the environment, from systems approach perspective, is only a theoretical idea without real implementations. Until now, the adaptation has relied on human experience and knowledge, and has therefore been highly subjective.

For adaptive production system, feedback loops and understanding the feedback are therefore essential. However, in case of static adaptation, the feedback and its processing don't need to happen in real time. The learning within production system can happen either on human or system level and this learning can turn into adaptivity of any of the types presented in the Figure 1.

2.3 Capabilities in adaptation

When adapting an existing production system for new product requirement two different approaches can be taken. According to Bi et al. (2008) the first approach is to design the new

system from scratch and then compare it with the existing system to establish the required changes. This is not very practical method, because it can lead to large and unnecessary changes. The second approach is to start with the original specification of the existing system and to change it until it fits the new requirements. (Bi et al., 2008.)

The adaptation methodology developed by the authors follows the second approach. It is based on comparing and matching the capabilities of the current system with the capability requirements set by the new product. Every device in the manufacturing environment has certain properties and behavior. Some of these properties and behaviors allow the device to perform a technical operation. All of the devices and their properties have certain ranges and constraints. They can be for example technical properties such as maximum torque of the spindle or velocity range of the moving axis, or environmental constraints like maximum allowed humidity and temperature. Automatic matching of available devices against product requirements requires formalized and structured representations of the functional capabilities and constraints of the devices.

However, presenting a simple capability of an individual resource is not enough when designing or adapting complete systems. Adaptation planning problem deals with a heterogenic system level, where a combination of different system levels can be recognized (see Figure 2a). When adapting a production system for a new product, there may be stations that could be utilized as they are without any mechanical changes. However, in order to detect that, there is a need for a formal description of the combined capabilities of the station composed of multiple devices. According to Ueda et al. (2001), the design of production systems follows emergent synthesis where the local interactions between the artifacts of the system form the global behaviour through bottom-up development to achieve the purpose of the whole system. Due to these local interactions, the combined behavior of resources is something else than the sum of the behavior of each individual resource. Consider for example the problem of combination of robot and gripper, illustrated in Figure 2b.

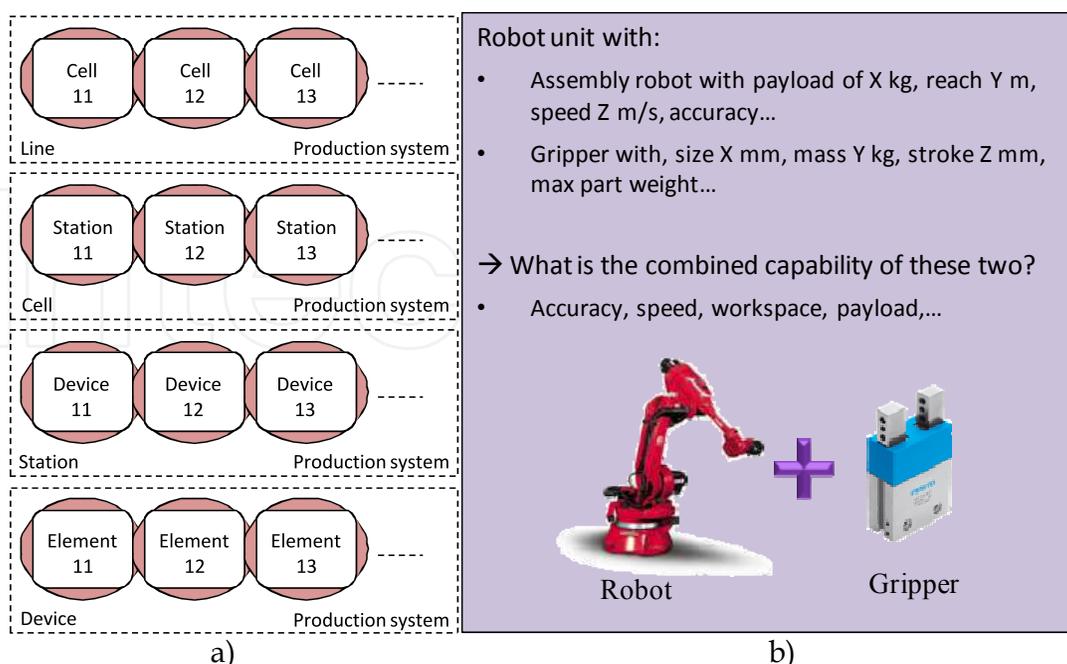


Fig. 2. a) Partonomy of different system levels; b) Example of combined capability problem (Järvenpää et al., 2011a).

A method to automatically derive the combined capability of multiple devices would not only enhance the adaptation, but also the original system design. This kind of model would allow the adaptation planning to start from the top, without considering each individual equipment inside the station. During the original system design it would allow to search for those resources, which would together fulfill the capability requirements, without the need to split up the capability requirements into most atomic pieces. In a holonic manufacturing environment (Nylund et al., 2008; Salminen et al., 2009) it would allow the capabilities of holarchies, composed of multiple individual holons, to be formulated based on the capabilities of the individuals.

3. Existing approaches for resource and capability descriptions

Usually, in order to describe production systems, different classification methods are used. Traditionally the devices are classified into groups based on their common properties or functions they provide, e.g. milling machines, lathes and so on. Unfortunately this kind of classification of systems is often limited. If the system is allowed to be member of one class only, then multifunctional system needs to be forced to the most appropriate class, even though it would have functionality fitting to multiple classes. One example of this kind of machine is a multifunctional universal CNC lathe, which is able to perform turning, milling and drilling. Depending on the context, the same device may be used to perform different activities. Therefore, this kind of classification doesn't provide enough expressiveness. To overcome this limitation, instead of classifying devices, the authors classify the functional capabilities that the devices provide. This way one device can have multiple capabilities to be used in different contexts and new capabilities can be assigned for the device when they emerge.

The manufacturing resource information and capability descriptions are considered as a fundamental basis for the various manufacturing activities including process planning, resource allocation, system and facility design as well as planning for system adaptation. By far there haven't been standardized information models to represent the manufacturing resource capabilities. Resource data models and tools for different system components exist, but they are vendor specific or very limited on their scope and not capable for describing the combined capabilities of multiple resources. In this section few existing approaches trying to overcome some of those limitations will be shortly reviewed.

In EUPASS project Skill concept was used to fill the gap between processes and equipment in the ontology. In the EUPASS ontology the skills were divided into basic skills and complex skills. The basic skills are the most fundamental skills, whereas the complex skills are combinations of more simple skills. (Lohse et al., 2008) Emplacement concept was developed to give a standardized description of the EUPASS modules comprising an assembly system (Siltala et al., 2008). Barata et al. (2008) presented a multiagent-based control architecture for shop floor system, where each agent representing a manufacturing component can be aggregated through the Broker Agent GUI to form a coalition of agents that coordinates higher level processes (complex skills) based on the ones available in its members. Based on Cândido and Barata (2007) ontology is used to identify which basic skills are necessary to provide complex skills. Unfortunately, there is no published material about how these two relating approaches solve the emergence of the atomic skills into complex ones, or how they are handling the complex skills within the ontology model.

Based on the example presented in Barata et al. (2008), the skill concept name is used to express the properties of the skill, which indicates that the skills don't have parameters, but just name. No reasoning seems to be done based on the technical properties of the devices. Therefore according to our best knowledge the problem has still remained unsolved.

The work performed at NIST (National Institute of Standards and Technology) by Ameri and Dutta (2008) aims to connect buyers and sellers of manufacturing services in web-based e-commerce environments. The matching is based on semantic similarities of supply and demand in terms of manufacturing capabilities. (Ameri & Dutta, 2008) Smale and Ratchev (2009) proposed a capability-based approach for multiple assembly system reconfiguration. Their work consists of capability taxonomy, capability model and reconfiguration methodology. The capability taxonomy is suited to both equipment specification and requirement definition, whereas the capability model combines the roles of the requirements definition, capability definition and capability comparison. (Smale & Ratchev, 2009.) The underlying problem to be solved in these two above mentioned approaches is somewhat similar to us - matching the existing capabilities with the required ones. However, neither of these approaches considers automatic capturing of the combined capabilities from the individual ones. The available system components are treated as individuals without considering their interfaces or co-operation.

Vichare et al. (2009) developed a Unified Manufacturing Resource Model (UMRM) to represent CNC machining systems and their auxiliary devices, such as workpiece and tool changing mechanisms, fixturing, robotic arms, conveyors, etc., aiming to capture information related to the manufacturing facility and its capabilities. UMRM is based on modeling kinematic chains of machines and is more concentrated on geometric aspects of the system. Therefore it is not adequate for describing the capability of e.g. an assembly system composed of multiple individual devices (not dividing the devices into elements like joints and axes).

4. Capability description method

The core of the developed adaptation planning methodology lies on the capability-based matching of product requirements and system capabilities. Automatic matching of available resources against product requirements requires formalized and structured representations of the functional capabilities, properties and constraints of the resources. This chapter will introduce a novel approach for describing and managing capabilities of manufacturing resources and combined capabilities of multiple co-operating resources with an ontology model. This modeling approach enables matching of products and resources based on their required and provided capabilities and this way supports rapid allocation of resources and adaptation of systems. First the definition of capabilities and combined capabilities will be given. Then the developed method for capability descriptions will be thoroughly discussed followed by the description of the components of the overall resource description.

4.1 Capabilities and combined capabilities

In the proposed approach capabilities are functionalities of resources, such as drilling, milling, moving and grasping (also called as capability concept name). Capabilities have parameters, which present the technical properties and constraints of resources, such as

speed, torque, payload, and so on. 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. For example capability with concept name 'moving', has parameters 'velocity' and 'acceleration'. 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, combined capabilities and competences. Combined capabilities are combinations of simple capabilities, usually formed by combination of devices, such as a robot gripper. Competences are human capabilities. Figure 3 represents the relations between capabilities, competences and combined capabilities.

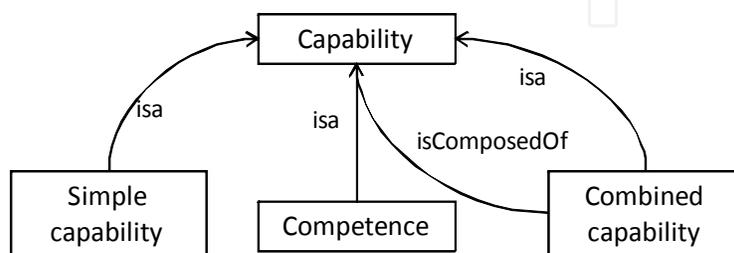


Fig. 3. Relations between capabilities, competences and combined capabilities (Järvenpää et al., 2011a).

There are two types of capabilities:

- **Strong capabilities** are those resource functionalities, which directly provide some kind of process, such as moving, grasping or releasing. Those resource characteristics that are directly related to some simple capability are given as the capability parameters.
- **Weak capabilities** are those properties and characteristics of a resource that do not naturally directly relate to any simple functional capability, but which are important when selecting the resource for a specific application. Weak capabilities are practically additional parameters for the other capabilities to aid in decision making, such as basic device information (containing device dimensions and weight).

4.2 Capability model

Ontologies play an important role in knowledge-based modeling approaches. They give a standardized way to present knowledge from different domains and knowledge sources. Lanz et al. (2008) used ontologies to structure the product, process and system related information, to include the meaning of the content to the models, and allow the information sharing between different applications. In the proposed approach, ontological modeling is used to represent the resource capabilities and constraints. The CoreOntology defined by Lanz (2010) is being used as a basis for describing the product, process and system related information. It allows basic descriptions relating to resources to be formalized. Now in the presented approach, it 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 is saved into a Knowledge Base (KB), described in detail in (Lanz, 2010).

The proposed capability modeling and matching method is based on capability modularization, enabling to build a dynamic link between product requirements and

production system. The capabilities are divided into simple and combined capabilities, like discussed in the previous section. The approach is based on functional decomposition of upper level capabilities (combined capabilities) into simple capabilities and assigning these simple capabilities for individual devices in a modular way. In many cases the functionalities of the systems can not be completely decoupled to certain single devices. However, in the presented approach the capabilities are artificially divided on the ontology level to support this capability modularity. When multiple devices are combined, the simple capabilities form combined capabilities. Upper part of the Figure 4b illustrates an example of such division by transportation capability in case of a robot unit consisting of a robot and gripper. The robot alone has only the ability to move its joints within a workspace. When combined with a suitable gripper, together they are able to transport pieces from one place to another.

The modeling of the combined capabilities is handled within the ontology using capability associations as rules how the combinations are formed. In the resource ontology, the devices are assigned the simple capabilities that they possess. 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 4a represents the metamodel for defining the combined capabilities. The same combined capability metamodel can be used in different domains. Definition of the domain specific capabilities, as well as the input and output associations for creating combined capabilities, require domain expert knowledge. For example in the context of this research, manufacturing engineering knowledge is required for the definition of the capabilities used in the case studies. Figure 4b represents the usage of the metamodel in the manufacturing domain.

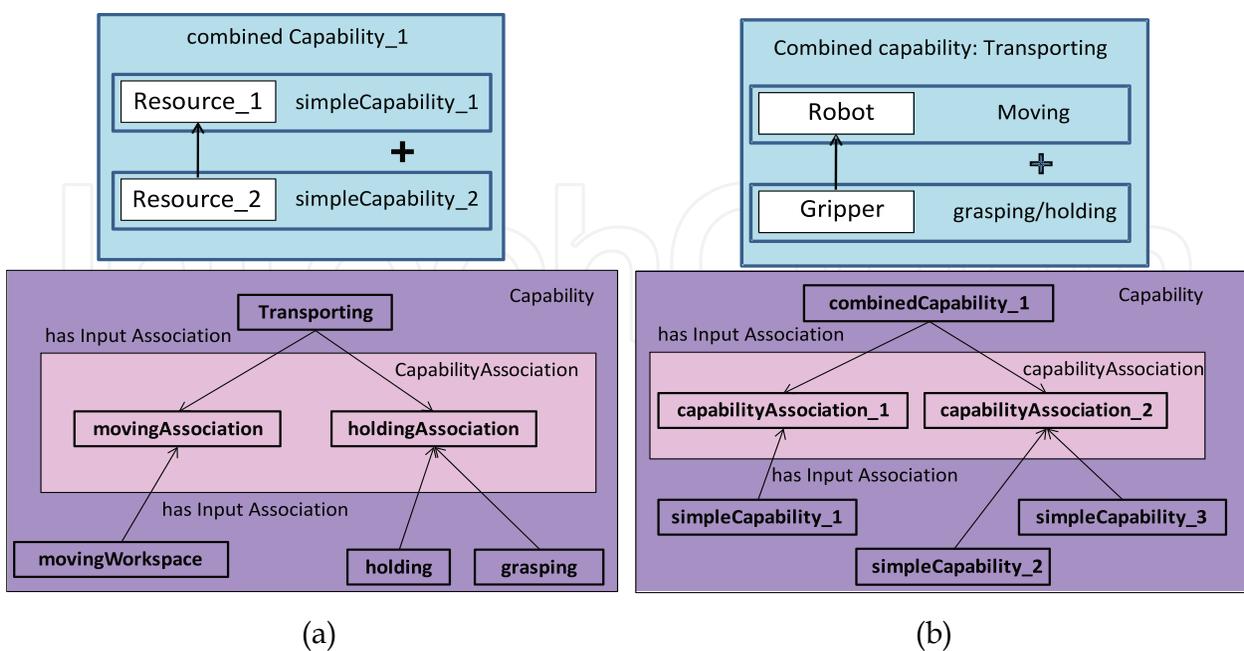


Fig. 4. a) Metamodel for the combined capabilities; b) Instantiated combined capability model.

The principles of the capability model are:

- Resources have simple capabilities, which provide some capability associations as their outputs.
- Combined capabilities require some capability associations as their inputs, e.g. in order to use the combinedCapability_1, both *capabilityAssociation_1* and *capabilityAssociation_2* have to be satisfied.
- Different simple capabilities can provide the same capability association as their output, like simpleCapability_2 and simpleCapability_3 in the Figure 4a.
- When a device or combination of devices provides output for all the required capabilityAssociations (*capabilityAssociation_1* and *capabilityAssociation_2*, in case of Figure 4a) the combined capability (combinedCapability_1) emerges.
- The capability model can have multiple levels, for example the combinedCapability_1 may provide an input association for some other combined capability. In this case the combinedCapability is treated as a simpleCapability from the upper level capability's point of view.
- The capability model can be extended freely upwards and downwards and new capabilities and capability associations can be added through learning processes. The detail level of the model can be further extended without disrupting the whole concept.

Capability model defines the generic capabilities, i.e. a pool of capabilities that can exist in a system. These generic capabilities are assigned to the resources and they become resource specific when filled with resource specific parameter values. With the presented capability model alone it is possible only to merge the capability concept names. On a more detailed level the parameters of the combined capabilities need to be defined, based on the parameters of the individual simple capabilities. The detailed level reasoning with the capability parameters will be handled by holonic reasoning. The holons use a rule base to define the parameters of the combined capabilities.

4.3 Components of the resource description

Manufacturing environment is constantly changing and the condition and capabilities of the resources change during their usage. Therefore it is important that the description of the resource is updated over time. For this reason the devices have two separate, but linked representations within the ontology: device blue prints and individual devices. The device blue print describes the capabilities and properties of one type of device, like given in suppliers' catalogues. The individual devices are presented in a separate class having a reference to the blue print device and presenting the properties of a particular individual device it is a representation of, i.e. individual resource existing on the factory floor. Figure 5a shows the relations between the device blue prints and individual devices.

The representation of individual devices holds the collected and measured lifecycle information of the device, which can be used in the planning process of reuse and adaptation. Collected raw lifecycle and history data is filtered and relevant key figures, such as Mean Time Between Failure (MTBF), Mean Time to Repair (MTTR), maintenance costs, reliability, operation time, estimated remaining lifetime, are calculated and saved as part of the resource description. Also the capabilities of the individual devices can be updated

based on the measured values from the factory floor, e.g. if the accuracy of the machine changes. The resource behavior log is constantly recording the operational information of the resources on the factory floor. The knowledge on how a specific resource or system did in a specific process while processing a specific part can later on be used for resource selection for similar products.

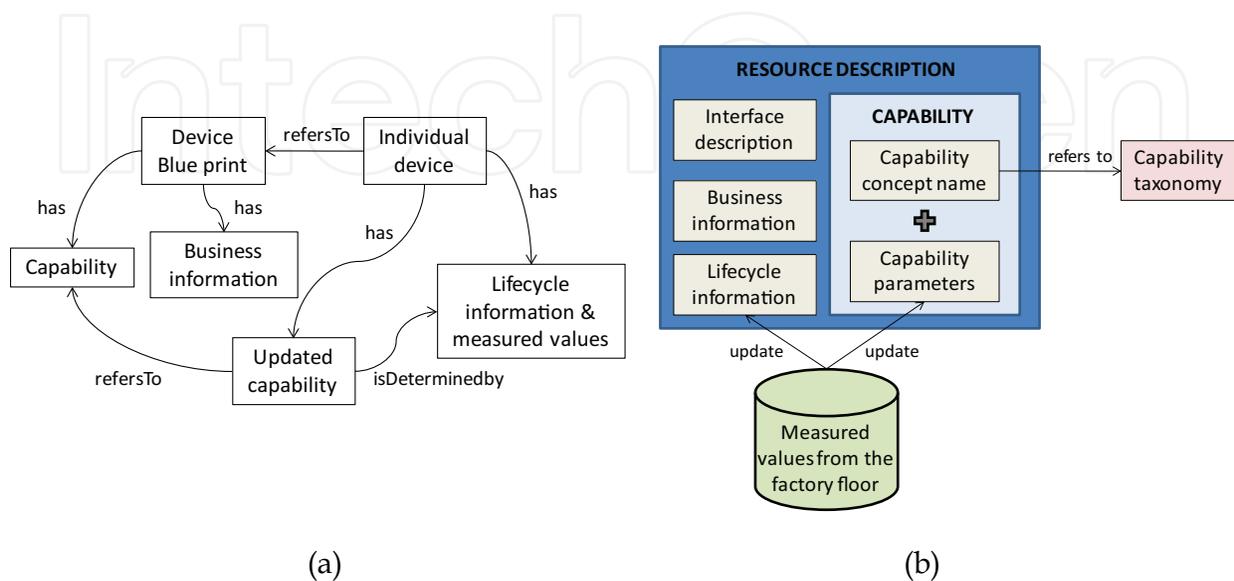


Fig. 5. a) Device blue prints and individual devices; b) Components of the resource description, modified from (Järvenpää et al. 2011a).

Figure 5b represents the components of the resource description. In addition to capability information containing the capability concept name and capability parameters, also collected lifecycle information and interface definition are part of the description. Also business information, such as purchase or rental costs can be added as part of the description to aid in decision making. The capabilities (only functional, i.e. strong capabilities) have a property of “hasCapabilityTaxonomy”, by which they refer to the correct level in the capability taxonomy, described more in the next section.

5. Capability based matching of product requirements and systems to support adaptation

The core of the developed adaptation planning methodology lies on the capability-based matching of product requirements and system capabilities. Figure 6 represents the reference architecture and explains the concept of the capability based mapping of product requirements and system capabilities. The mapping is based on capability taxonomy connecting the product and resource domains together. Taxonomy, included into the CoreOntology, is used to make a crude search that matches the resources with required capabilities. The detailed reasoning with the combined capabilities and their parameters is based on holonic negotiation process. Next sections will first discuss about the definition of product requirements and then continuing with the requirement-capability matching based on the ontology definitions and holonic reasoning, as shown in the reference architecture.

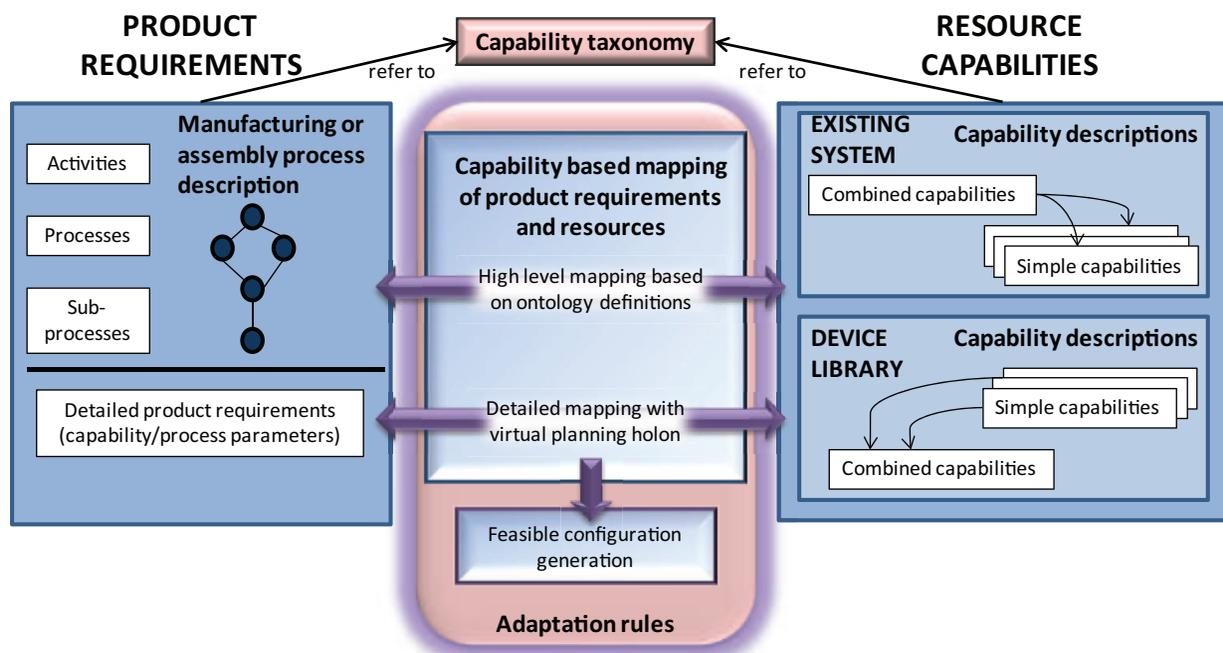


Fig. 6. Reference architecture for capability-based mapping, modified from (Järvenpää et al, 2011a).

5.1 Definition of the product requirements

Product requirements are those product characteristics or features which require a set of processes in order for the product to transform towards the finished product. These processes are executed by the devices and combinations of devices possessing adequate functional capabilities. The product requirements can be expressed by the required capabilities and their temporal and logical order. (Järvenpää et al. 2011b.) In the proposed approach the product requirements are expressed in the form of a pre-process plan, generated by a tool called Pro-FMA Extended (Garcia et al., 2011). Pro-FMA Extended is a software tool, which recognizes the product features from VRML or X3D model and generates the high-level process plan, pre-process plan, based on those features. Basically, the pre-process plan is a generic recipe on how to manufacture the part or product. Each feature contains its characteristics - shape, type, material, tolerance and geometric dimensions - based on which the pre-process plan can be generated. The pre-process plan is an ordered graph of generic activities referring to specific levels on the capability taxonomy stored in the Knowledge Base. (Garcia et al., 2011.)

5.2 Reasoning about the capabilities based on ontology definitions

Capability taxonomy, integrated to the CoreOntology allows, together with the developed capability model, matching between product requirements and resource capabilities on different levels of detail. Like discussed in the previous chapter, the activities in the pre-process plan of the product refer to specific levels on the capability taxonomy. Naturally also the capability instances refer to a certain level in the capability taxonomy enabling the link between products and resources providing the capabilities. See a simplified example of the capability taxonomy in Fig. 7. Due to the limited space, only a small part of the taxonomy with some examples is presented.

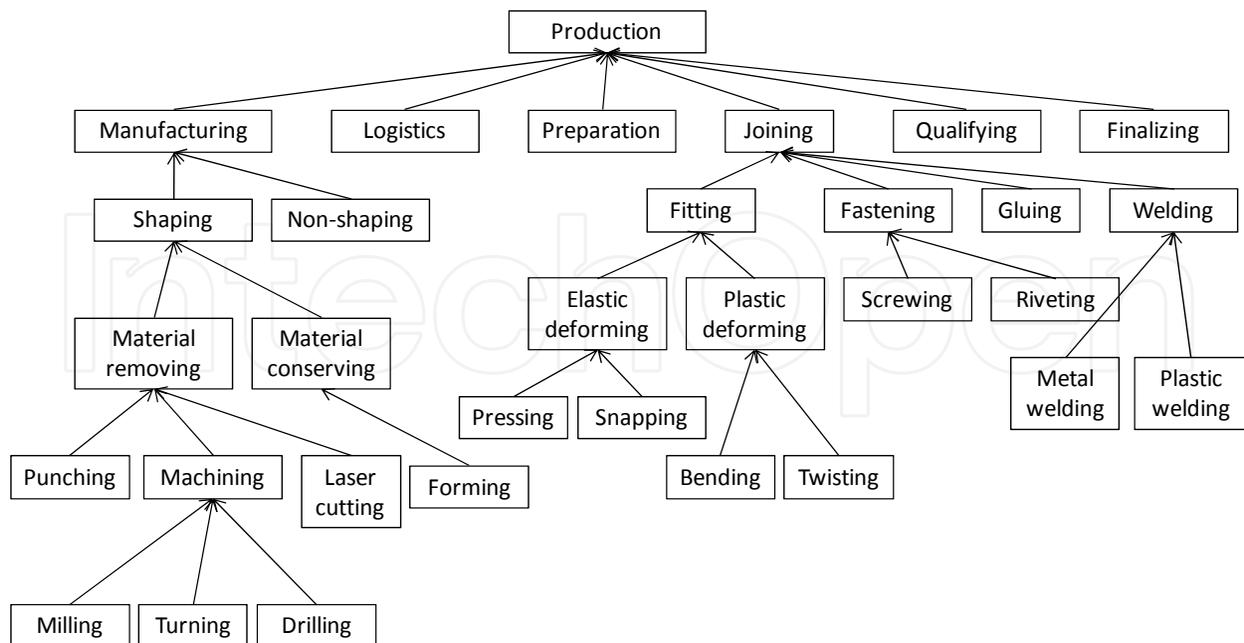


Fig. 7. Simplified capability taxonomy (Järvenpää et al. 2011a).

The taxonomy level where the pre-process plan is referring depends on how detailed information about the required or desired processing methods is available. For example, the product designer may have defined that a specific joining method, such as riveting, should be used to join two parts together or she/he may have only defined that some sort of joining capability is required, leaving possibility to determine the joining method based on the capabilities available on the factory floor. In the first case the product requirement is directed on the particular method in the capability taxonomy, whereas in the latter case the requirement is directed on the joining level in the taxonomy.

The taxonomy allows to search for different devices that are able to perform the same function (e.g. material removing) by different behavior (e.g. milling, turning, drilling,...). The parameters of the capabilities will then determine if the suggested device is able to fulfill the given requirements. For example if the requirement is [material removing, hole of diameter 20mm and depth 50mm, aluminum], the parameters of the capabilities which are subset of the material removing capability (e.g. milling, turning and drilling) will then express, which device combination is able to provide required material removing with required parameters.

As the devices are assigned the simple capabilities they possess, based on the defined capability associations, the device combinations, contributing to certain combined capability, can be identified and queried with SPARQL RDF query language. Similarly it is possible to reason out the capabilities that the resource combinations have. Fig. 8 illustrates how the capability associations (inputs and outputs) are used to make a match between the capabilities existing with the current resources and the required capabilities. By using the associations it is possible to answer e.g. to following questions: "Which devices I need to combine in order to get a certain combined capability?" "What combined capabilities a certain combination of devices can have?"

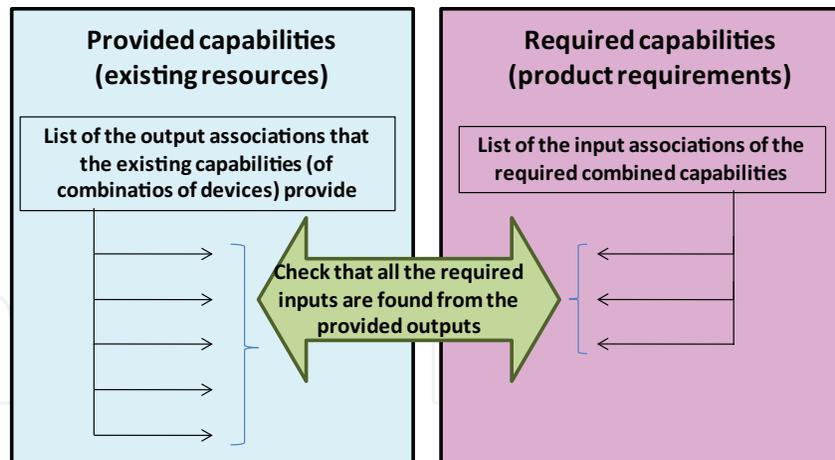


Fig. 8. Matching of capability output and input associations.

Figure 9 gives an example of the matching. The input and output associations are written with italic (e.g. *spinningTool*). As seen in the figure the combined capabilities are formed by hierarchical climbing from lower level combinations to up, e.g. the tool holder and threading cutter combination is considered as threading tool on the next level. Queries are implemented to the Knowledge Base as services that can be called by different applications.

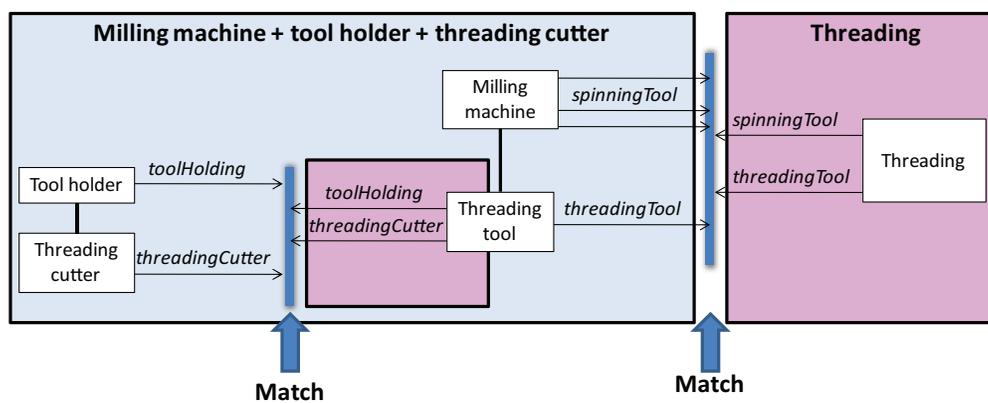


Fig. 9. Example for matching the capability input and output associations.

Summarized, the ontology serves as the representation of taxonomic (is-a) and partonomic (is-part-of) relations between capabilities. The partonomic relations are enabled by the use of the capability associations. As the capability associations are only able to identify the concept name of the combined capability, more intelligent reasoning is required for defining the combined capability parameters. For this purpose, holonic reasoning is applied.

5.3 Holonic reasoning about the capabilities

Ontology is used to make a crude search that matches the resources with required capabilities. The detailed reasoning with the capabilities and their parameters is based on holonic negotiation process. Each different machine on the factory floor has its holon representation that presents that specific machine in a digital world and gets the capability information from the Knowledge Base. When the preliminary matches are found using ontology queries, these scenarios are tested with each holon to make sure that it can actually produce the item in

question. This reasoning will output a list of machine-capability pairs that match the request at hand. Since the holons gather the actual lifecycle data of the machines, it can be used in the decision making process. For example, if the accuracy of the machine has not been as good as promised in the data provided by the manufacturer, the holon can adapt new values from the history data and use those for the reasoning.

The holonic reasoning requires rules in which the reasoning can be based on. The rule-base is currently under development, consisting of three types of rules:

Combined capability rules:

- Rules, which determine how the parameters of combined capabilities can be formed from the parameters of individual capabilities. The aim is not to provide detailed analysis of e.g. workspace or kinematics of the device, but to perform scenario modeling of possibly suitable devices and device combinations. For kinematics and detailed workspace definitions virtual simulation tools are used to validate the results got from the reasoning based on the digital information.
- Example rule: When robot and gripper are combined, the payload of the combination is robot payload minus the gripper weight or gripper payload if it is smaller than the previous difference.

Domain expert rules:

- Rules, which define how the capability and its parameter information are applied in each domain when matching with the product requirements. These rules include, e.g. in machining process domain, how the achieved feature depends on the tool shape and type. Our focus is not, however, on defining the domain specific rules in detail, but to demonstrate the concept of how to use those rules and capability definitions in production system design and adaptation in case of changing product requirements and other external and internal changes.
- Example rule: In milling the nose radius of the cutter has to be same as the required rounding inside the machined pocket.

Adaptation planning rules:

- Rules defining how other criteria, such as availability and scheduling, device condition and lifecycle, as well as user and company specific criteria relating for example to costs, eco-efficiency or speed, is used in the final resource selection and configuration generation. These rules are usually given by the user in a specific case and are therefore dynamic in their nature.
- Example rule: If the amount of ordered items is 20 or more, use the fastest machine for the product manufacturing. Otherwise, use the cheapest option.

Despite the generation of rule-base allowing automatic reasoning, human intervention is still required. The goal of the holonic system is not to make everything automatic, but to keep the human involved in the decision making and control loop. In the presented approach it means validation of the automatically generated scenarios and selection of the most desirable solution based on the user specific criteria. Positive aspect with the rule-base is, that it can be extended incrementally on the fly leading to more and more accurate and realistic reasoning results requiring less human intervention.

6. Case implementation for dynamic adaptation

In the context of this research, a dynamic operation environment consisting of the hardware in the TUT machining laboratory environment and modular ICT architecture has been implemented. The build environment utilizes holonic manufacturing paradigm and integrates existing technologies resulted from different projects into one operation environment. The adaptivity of this holonic system rests on SOA (Service Oriented Architecture) -based communication and negotiation between entities through open interfaces, and matching of resource capabilities against product requirements.

The main character of the built holonic system is that the status of the production system and desired goal (defined as order connected to product model) are known, but the steps for reaching the goal, in this case the routing order of the parts in the factory floor, is not predefined. The holonic system follows service oriented architecture (SOA), where the resources provide services through their capabilities. When an order enters to the holonic framework, the system will search for those resources, which can alone, or with some other resource, satisfy the requested service. The holons will then negotiate to determine the best resource for the given situation or the part is directed to first available resource combination that has a capability to produce the part or a specific feature. (Järvenpää et al. 2011c.)

This section will give a detailed description of the implementation of the dynamic operation environment utilizing the capability descriptions. The implementation was built into the Academic Research Environment of TUT. This section will first shortly introduce the hardware part of the environment and then concentrate on the developed modular software system architecture. After those, the process and information flows in the dynamic operation environment will be thoroughly discussed.

6.1 Hardware in the dynamic operation environment

The hardware part of the research environment consists of several manufacturing resources and work pieces as physical manufacturing entities, the real parts (see Fig. 10). Each of them has their corresponding computer models and simulation environments as their virtual parts. The resources of the research environment, offering different manufacturing capabilities, are:

- Machine tools (a lathe and a machining centre) for machining operations
- Robots for material handling and robotized machining operations
- Laser devices for machining, marking and surface treatment
- An automated storage for storing blank parts and finished work pieces
- A punch press, existing only virtually, for the punching of sheet metal parts.

The work pieces are fairly simple cubical, cylindrical, and flat parts in shape. They have several features with parameters that can be altered, such as part dimensions (width, length, and depth), number of holes, internal corner radiuses, sheet thickness as well as material and tolerance requirements of the finished products. These features determine the product requirements for which suitable capabilities need to be found. The system in control level does not distinguish the tasks needed to be done, but the tasks are purely assigned based on the capability requirements.



Fig. 10. Resources on the factory floor and manufactured products.

6.2 Modular ICT architecture

The software system architecture, illustrated in Fig. 11, has several different interoperating software modules each providing one or two essential functions for the whole holonic manufacturing system. The architecture follows the dynamic modularization principles being designed in such way that each of the modules can be replaced with a new module if needed without disturbing the whole system. The interoperation of the modules is mainly based on the shared information model and common knowledge representation, the Core Ontology, and modular services. Each of these modules requires specific domain related information and by processing the information they provide a set of services.

Figure 11 shows the simplified information flows between the modules forming the ICT architecture. The communication between the Knowledge Base and the modules is done by RDF and XML messages (depending on the situation) using SOAP. The communication between the DeMO tool, UI/control holon and the machine UIs is done by using XML-RPC calls.

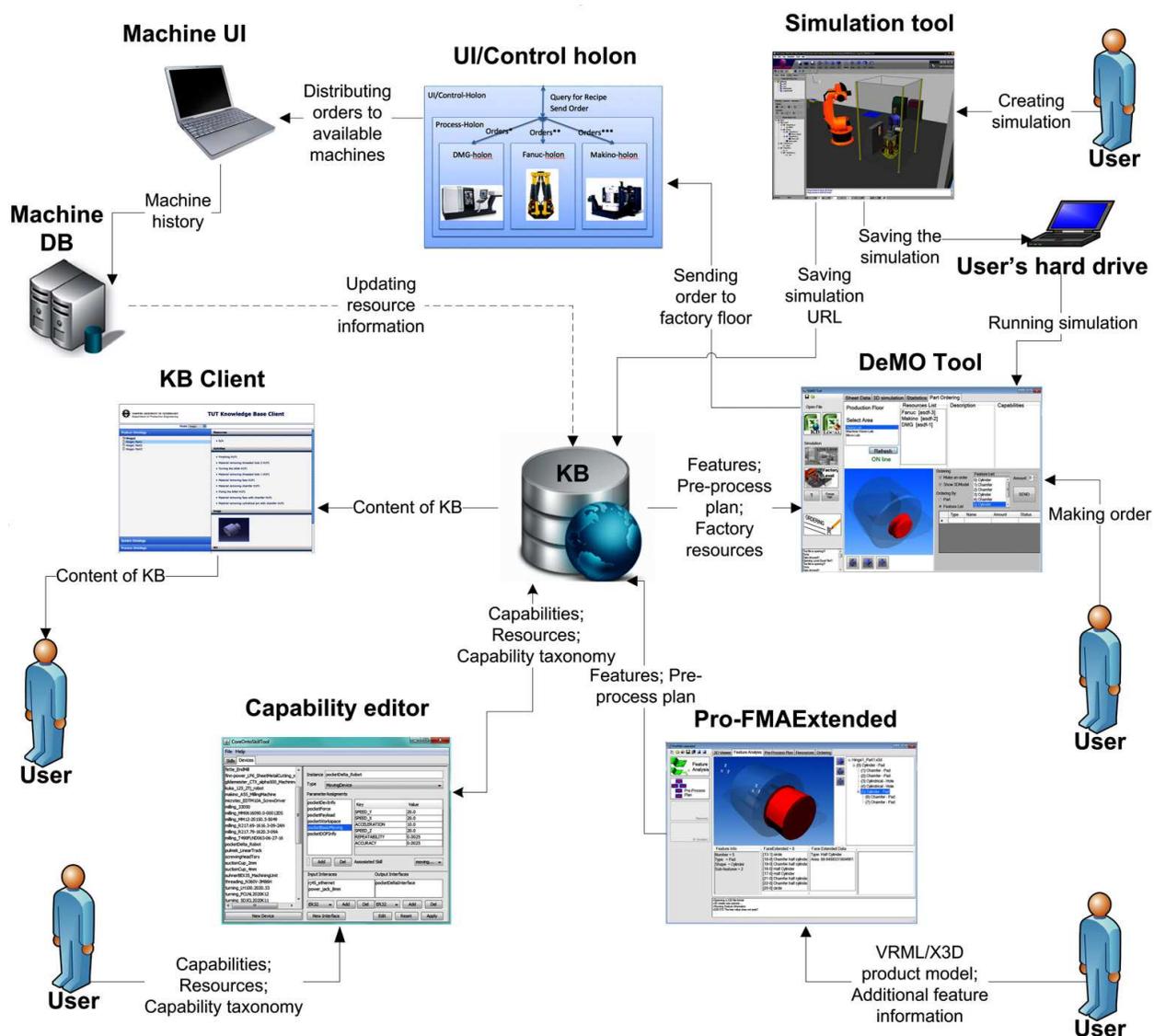


Fig. 11. Modular ICT architecture and simplified information flows during the operation of the dynamic operation environment.

In the following is listed the roles of different ICT tools in the dynamic operation environment:

- **Pro-FMA Extended** is used to define the product requirements from the product model given in VRML or X3D format (Garcia et al., 2011).
- **Capability Editor** allows user to add resources to the ontology and assign them capabilities and capability parameters. It is also used for adding new generic capabilities to the KB, as well as for creating associations between the capabilities. (Järvenpää et al. 2011c.)
- **Simulation tool** is used for verification and creation of the manufacturing or assembly scenarios.
- **Decision Making and Ordering Tool (DeMO tool)** is used for setting up the orders in this environment. The main function of the DeMO tool is to verify connection to factory floor and forward the orders to the holonic UI.

- **UI/Control holon** manages the process flow and distribution of tasks to each manufacturing or assembly cell. This system distributes the tasks to suitable and available cells or stations based on the capability requirements defined by Pro-FMA Extended.
- **The Knowledge Base (KB)** store the information created by Pro-FMA, Capability Editor and DeMO tool in a form of common knowledge representation, Core Ontology (Lanz, 2010). This system serves also as a reference architecture, since it can handle closed models as references.
- **Web-based KB client** is used for human friendly knowledge browsing and content verification. This tool serves as online product data management (PDM) user interface (UI).

6.3 Information and process flows during a case scenario

This section will explain step-by-step the activities taken when new product comes to the production in the developed dynamic operation environment. It will also explain how new resources and new capabilities are integrated into the system on a digital level. Fig. 11 gave an overall view of the information flows in the developed dynamic operation environment. Fig. 12 represents the flow of the activities and reasoning process in the environment when all the available resources are already described in the Knowledge Base. Before discussing in more detail those activities shown in the figure, the resource description activities will be explained. This means that in the next process description, the phases a and b are not shown in the graph below. The process description is divided into activities on digital, virtual and real levels.

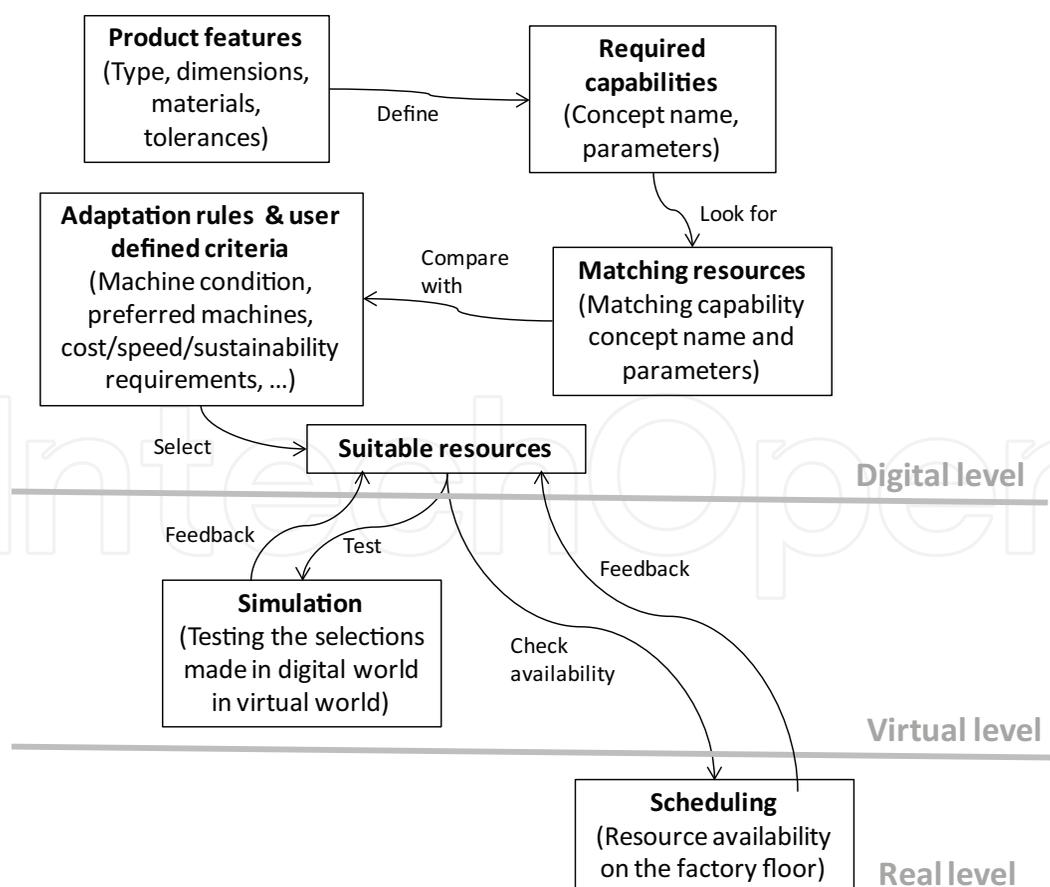


Fig. 12. Process flow in the holonic manufacturing system framework.

Activities on a digital level

a. Adding new resources to the Knowledge Base

New resources are added to the Knowledge Base with the Capability Editor. In the Capability Editor, the user can specify the name and type of the resource and describe its mechanical, control and energy related interfaces. The main task when describing the resources is to assign them capabilities. The user will select the pre-defined simple generic capabilities from the Capability Editor and define the resource specific parameters for the capability, making the capabilities unique for each resource. If a suitable generic capability is not pre-defined, a new capability has to be defined first and then assigned to the resource (see the phase b). Fig. 13 shows the user interface of the Capability Editor for adding resources to the KB. In the ParameterAssignments box, the resource specific instances of the generic capabilities are created and parameter values given.

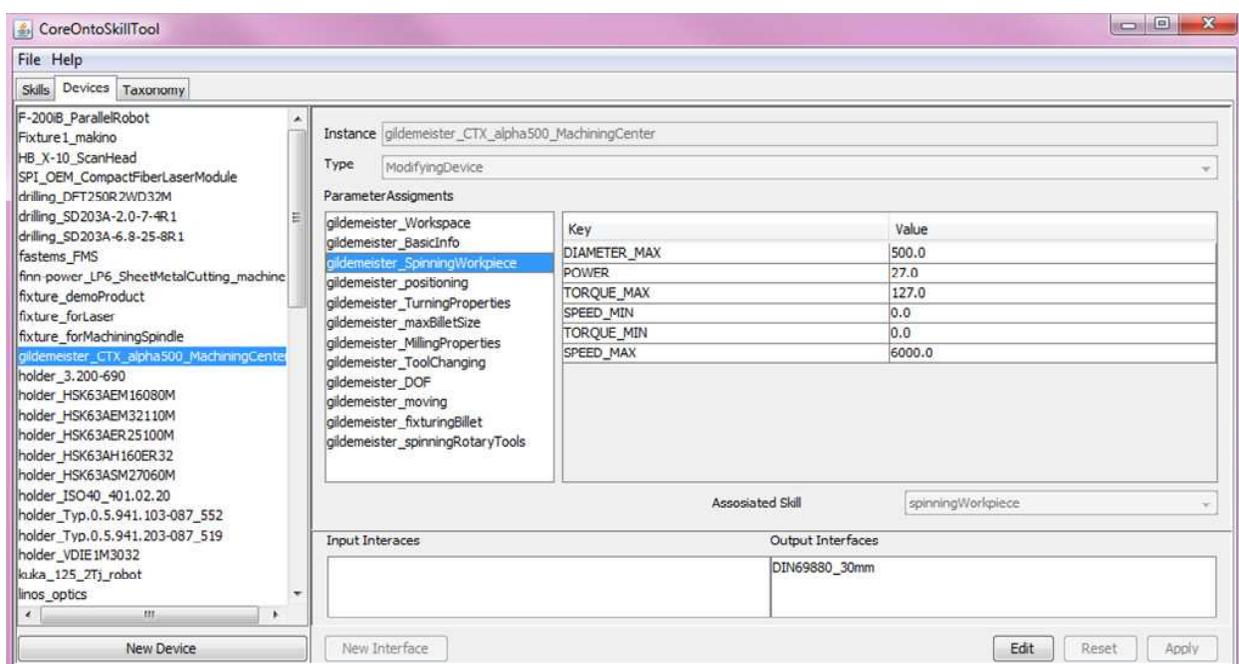


Fig. 13. Capability Editor – Adding resources and capabilities to the Knowledge Base.

b. Adding new generic capabilities to the Knowledge base

If suitable generic capabilities are not pre-defined in the KB, the user needs to first define those with the Capability Editor. Definition of the generic capabilities includes definition of the capability concept name and capability parameters. In case of the strong (functional) capabilities, also the link to the capability taxonomy will be created. Figure 14 shows the user interface for adding capabilities in the Capability editor.

1. Definition of product features

The product model is first sent in VRML or X3D format to Pro-FMA Extended software, which will recognize the features that need to be manufactured. It recognizes the shape, type and dimensions of the feature. The user can manually give the material and tolerance information, as well as special finishing information for the features. This feature information is then sent to the Knowledge Base for further use. The upper part of the Figure 15 presents the user interface of the Pro-FMA Extended with the recognized features and their parameters.

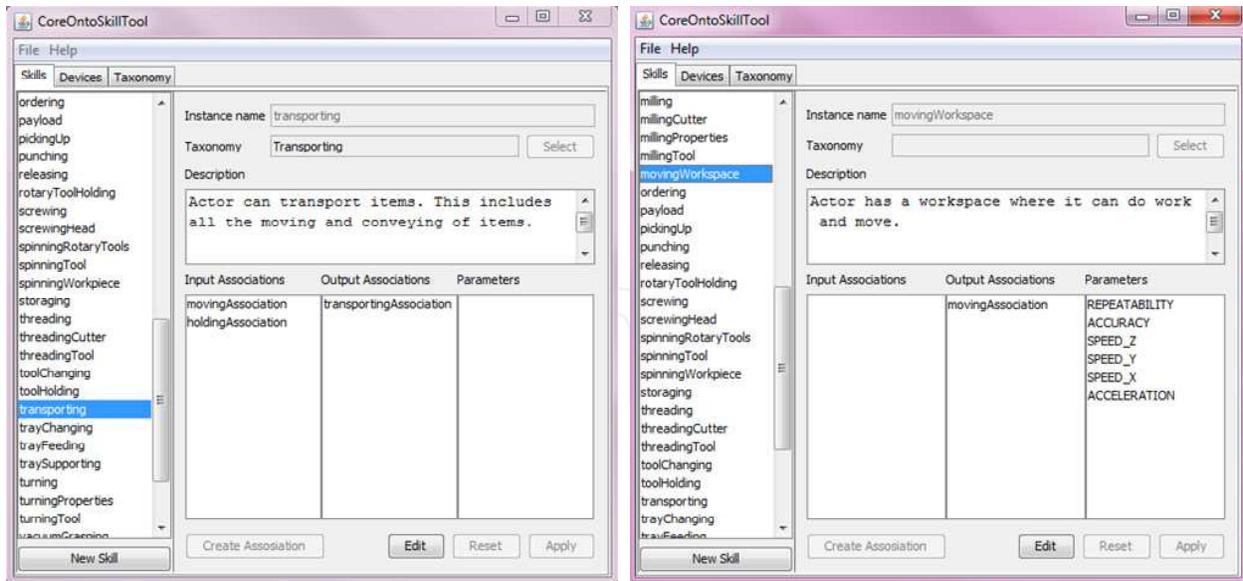


Fig. 14. Capability Editor – Adding new capabilities to the Knowledge Base.

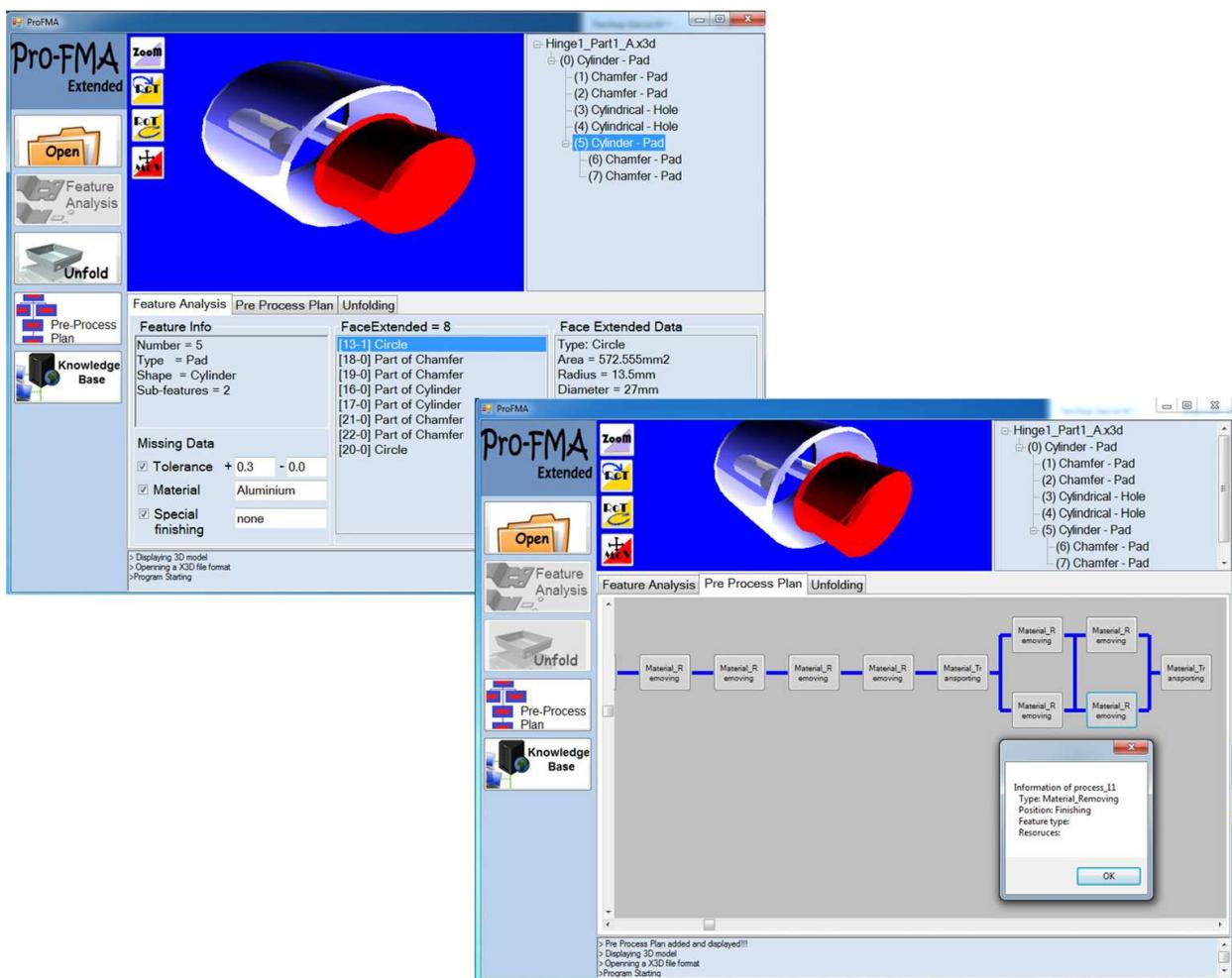


Fig. 15. Pro-FMA Extended – analyzing the product features and creating the pre-process plan, modified from (Garcia et al., 2011).

2. Definition of pre-process plan

Based on the recognized features the Pro-FMA Extended defines a pre-process plan for the part/product manufacturing (see lower part of the Figure 15). The pre-process plan is created based on the rules embedded into the software. The pre-process plan determines the processes on high level (e.g. material removing, material adding) leaving possibility to use different methods (capabilities) for manufacturing the part/product. The pre-process plan is then sent to the Knowledge Base, where the steps of the pre-process plan are linked with the generic capabilities in the capability taxonomy. Pre-process plan together with the feature recognition data defines the capability requirements for the part/product manufacturing in the following way. Generic process names in the pre-process plan refer to certain capabilities in the capability taxonomy. Each step in the graph represents one generic capability requirement. The capability parameters relating to the product features, e.g. size and material requirements remains in the product side in the ontology. Other capability parameters, e.g. required speed, relate to the order specific parameters, like production volume and delivery date. These are defined in later phase when creating the order with DeMO Tool.

3. Matching the capability requirements and resources

The matching of the required capabilities and suitable resources is done based on the capability taxonomy. Both the pre-process plan and the resource specific capabilities have a reference to the capability taxonomy allowing the matching. Definition of combined capabilities is implemented into the KB as services, i.e. other software can use those services to determine the capabilities of device combinations and to search for suitable devices.

4. Applying context specific criteria and rules for the resource selection

The DeMO Tool (Decision Making and Ordering Tool) is used to place the order and send it to the factory floor, as well as to select the preferred resources and apply user defined criteria. First the product to be ordered is selected. The feature information related to that specific product is retrieved from the Knowledge Base. The different manufacturing stations existing currently on the factory floor and defined in the KB are shown in the DeMO tool. While selecting one station, the tool shows the individual machines and tools comprising the station. It also shows the capabilities and combined capabilities of the selected resources. By comparing the required capabilities and provided capabilities the matching resources can be identified. This limits the options of useable stations. Based on the context specific criteria (including adaptation and user specific criteria) the user can then further limit the amount of suitable resources by selecting the most desirable stations. The criteria can relate e.g. to sustainable performance metrics, machine condition, or speed requirements. Figure 16 shows the user interface of the DeMO Tool when sending the order to the specific machine on the factory floor.

Activities on a virtual level

5. Test manufacturing by simulations

With DeMO tool, it is possible to run pre-created simulations of the processes and view the statistics before placing the orders. Because it is very difficult to determine some capabilities, e.g. workspace of the device combination, accurately on a digital level, simulations may be required to validate the feasibility of the matched capabilities.

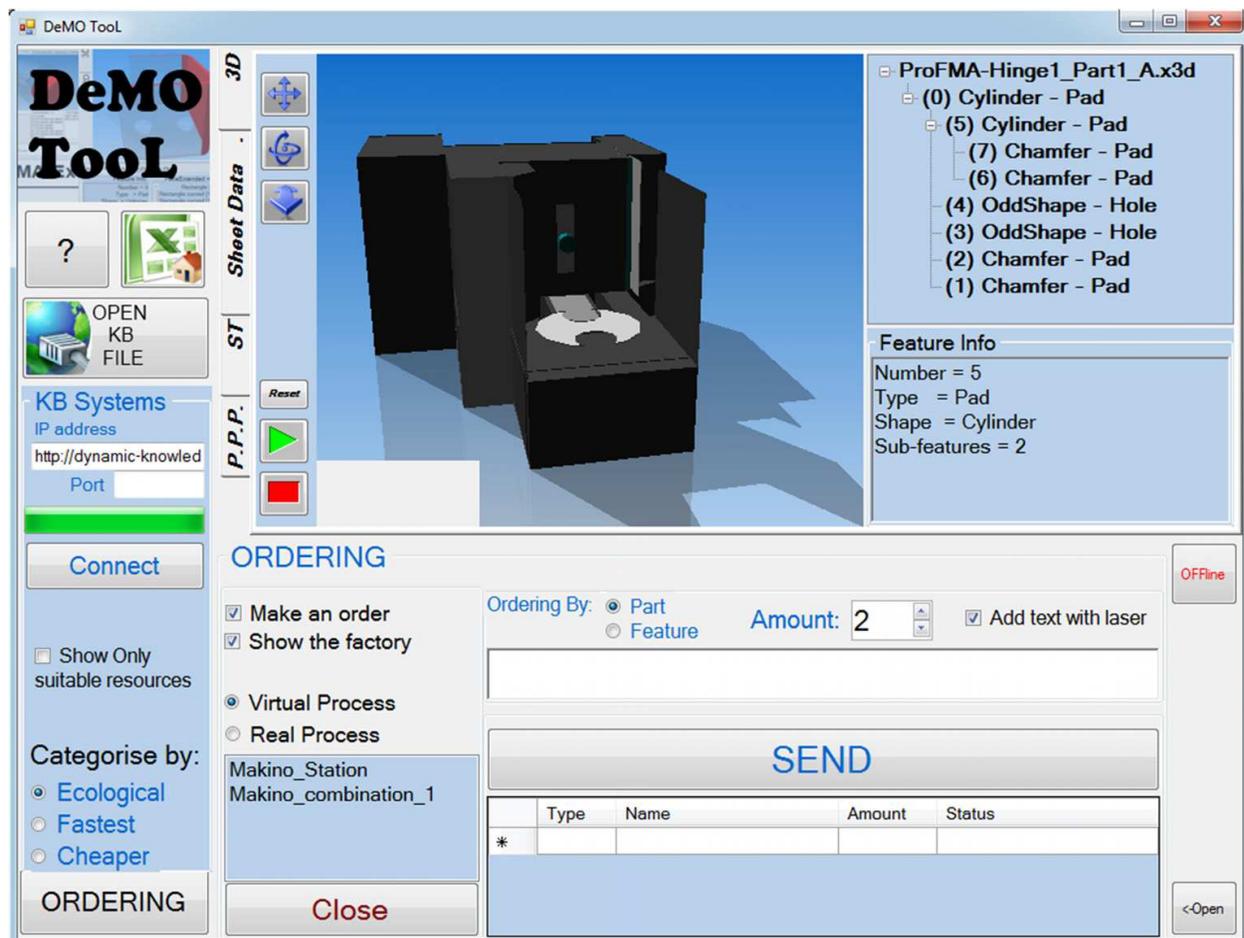


Fig. 16. DeMO Tool – Sending order to the factory floor.

Activities on a real level

6. Checking the availability of resources and routing the order

When the order is placed by the user, the UI/Control holon (scheduling holon) will check the availability and status of the suitable resources on the factory floor by discussing with the machine UIs and then communicate the information with the DeMO Tool. The order is then routed to the first available and suitable manufacturing station (taking into consideration the context specific criteria). The Machine database will save the resource behavior log information, including the status of the resource, completed orders, possible measured values, and so on, which can later on be used for decision making.

7. Future work

In order to make the concept and operation of the dynamic operation environment more intelligent and autonomous, some future developments have been envisioned for the tools forming the environment.

Capability Editor: Currently the Capability Editor is used to manage only the device Blue Print information and the individual devices and device combinations need to be created with the Protégé ontology editor. However, because Protégé is not an optimal tool for

manipulating large amounts of data, this function will be later on implemented to the Capability Editor.

Pro-FMA Extended: Currently the pre-process plan defines only the very high level capability requirements, e.g. material removing or material adding. In the future the algorithms for the pre-process planning will be further developed to enable more intelligent reasoning based on the recognized features and the additional user given information.

DeMO Tool: Currently the DeMO tool shows only those device combinations that exist on the factory floor (i.e. those, which are created to the KB). Later on it should be possible to display all the possible device combinations having the required capability (also those, which are not existing currently on the factory floor). If suitable capabilities don't exist currently on the factory floor, new combinations of existing devices would be created and shown in the DeMO Tool. The DeMO Tool would discuss with the UI/Control holon about the availability of the devices. Only available devices can make new combinations.

Usage of history data: The history data collected by the Machine DB can be later on used by the UI/control holon e.g. to evaluate how well the resource performed in different situations. This information can then be utilized when resources for similar applications are needed. The history data can also be used for updating the capability information in the Resource KB, e.g. the accuracy. The history data should be handled by the role engine, so that it could be associated with the specific roles that were used while collecting the data. This way the content and context information could be connected enabling knowledge to emerge. This knowledge can then be utilized for successful adaptation.

Rule-base for the holonic reasoning: The development of the rule-base for detailed reasoning about the capabilities has been started, but it is not yet implemented as a part of the system. The implementation of the rule-base will allow more automatic reasoning and more accurate results requiring less human intervention.

8. Conclusions

The operation and business environment changes rapidly. Ability to quickly adapt itself to new requirements has become a crucial enabler for the industry to gain operational flexibility. Support for adaptation is required from all operation levels. A crucial enabler for this kind of dynamic operation environment is modular ICT architecture following the holonic principles. This chapter presented a concept and case implementation of a new kind of dynamic operation environment based on holonic framework. The presented approach enables the step towards more intelligent and adaptive production systems by applying four technical solutions: service oriented architecture allowing the customers to place their orders and resources to advertise their capabilities; open interfaces enabling interoperability; common language and structure for the communication based on the ontology; and holonic negotiation process allowing to make the match between requests and offerings and utilize other criteria for the final decision making.

The introduced capability description method is a crucial enabler for the operation of the presented dynamic operation environment. Formalized capability descriptions allow the holons to advertise their capabilities to other holons in the system, and to autonomously organize the production based on the available capabilities. Capability descriptions allow also

automatic methods to find suitable system components and to build alternative scenarios for different product requirements. As the matching based on the capability descriptions can not assure optimal accuracy of found solutions, human intervention is still needed to check the feasibility of the generated scenarios and to select the best one for the given situation.

The presented implementation of the dynamic operation environment provides information of the content and context of a manufacturing and assembly system. However, for achieving real intelligence and adaptivity, the past versus present status needs to be taken into consideration. In the current implementation this comparison is not yet done. More research actions towards a “learning factory” is required in the future.

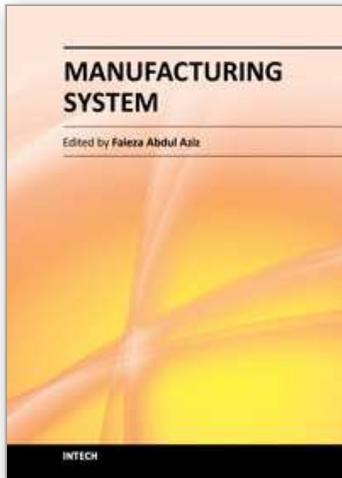
9. References

- Ameri, F. & Dutta, D. (2008). A Matchmaking Methodology for Supply Chain Deployment in Distributed Manufacturing Environments. *Journal of Computing and Information Science in Engineering*, Vol. 8, No. 1, p. 9, ISSN 1530-9827.
- Barata, J, Camarinhamatos, L. & Candido, G. (2008). A multiagent-based control system applied to an educational shop floor. *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 5, pp. 597-605, ISSN 0736-5845.
- Bi, Z.M., Lang, S.Y.T., Shen, W. & Wang, L. (2008). Reconfigurable manufacturing systems: the state of the art. *International Journal of Production Research*, Vol. 46, No. 4, pp. 967-992, ISSN 0020-7543.
- Bourguine, P. & Johnson, J. (Eds) (2006). *Living roadmap for complex systems science*, Open Network of Centres of Excellence in Complex Systems, 71 p.
- Cândido, G. & Barata, J. (2007). A multiagent control system for shop floor assembly, In: *Holonic and Multi-Agent Systems for Manufacturing*, A. W. Marik, V, Vyatkin, V. Colombo (Eds.), pp. 293-302, Springer, ISBN 3-540-74478-9.
- ElMaraghy, H.A. (Ed.) (2009). *Changeable and Reconfigurable Manufacturing Systems*, Springer, ISBN 978-1-84882-066-1, London.
- ElMaraghy, H. A. (2006). Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, Vol. 17, No. 4, pp. 261-276, ISSN 0920-6299.
- Garcia, F., Lanz, M., Järvenpää, E. & Tuokko, R. (2011). Process planning based on feature recognition method, *Proceedings of IEEE International Symposium on Assembly and Manufacturing (ISAM2011)*, ISBN 978-1-61284-343-8, 25-27 May, 2011, Tampere, Finland.
- Harms, R., Fleschutz, T. & Seliger, G. (2008). Knowledge Based Approach to Assembly System Reuse, *Proceedings of ASME 2008 9th Biennial Conference on Engineering Systems Design and Analysis (ESDA2008)*, pp. 295-302, ISBN 978-0-7918-4835-7, 7-9 July, 2008, Haifa, Israel.
- Jovane, F., Westkämper, E. & Williams, D. (2009). *The ManuFuture Road – Towards Competitive and Sustainable High-Adding-Value Manufacturing*. Springer, ISBN 978-3-540-77011-4.
- Järvenpää, E., Luostarinen, P., Lanz, M. & Tuokko, R. (2011a). Presenting capabilities of resources and resource combinations to support production system adaptation, *Proceedings of IEEE International Symposium on Assembly and Manufacturing (ISAM2011)*, ISBN 978-1-61284-343-8, 25-27 May, 2011, Tampere, Finland.

- Järvenpää, E., Lanz, M., Siltala, N. & Tuokko, R. (2011b). Activities and Corresponding Information Flows for Production System Adaptation Planning, *Proceedings of the 4th International Swedish Production Symposium, SPS11*, pp. 124-133, 3-5 May, 2011, Lund, Sweden.
- Järvenpää, E., Luostarinen, P., Lanz, M., Garcia, F. & Tuokko, R. (2011c). Dynamic operation environment – Towards intelligent adaptive production systems, *Proceedings of IEEE International Symposium on Assembly and Manufacturing (ISAM2011)*, ISBN 978-1-61284-343-8, 25-27 May, 2011, Tampere, Finland.
- Koren, Y. (2006). General RMS Characteristics, Comparison with Dedicated and Flexible Systems, In: *Reconfigurable Manufacturing Systems and Transformable Factories*, A. I. Dashchenko (Ed.), pp. 27-46, Springer, ISBN 3-540-29391-4.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., & Van Brussel, H. (1999). Reconfigurable Manufacturing Systems. *CIRP Annals - Manufacturing Technology*, Vol. 48, No. 2, pp. 527-540, ISSN 0007-8506.
- Lanz, M. (2010). Logical and Semantic Foundations of Knowledge Representation for Assembly and Manufacturing Processes. *PhD Thesis*, Tampere University of Technology, ISBN 9789-52152-3939, Tampere, Finland, p.138.
- Lanz, M., Kallela, T., Velez, G. & Tuokko, R. (2008). Product, process and system ontologies and knowledge base for managing knowledge between different clients, *Proceedings of the IEEE International Conference on Distributed Human-Machine Systems (DHMS2008)*, ISBN 978-80-01-04028-7, pp. 508-513, 9-12 March, 2008, Athens, Greece.
- Lohse, N., Maraldo, T. & Barata, J. (2008). EUPASS std-0007: Assembling Process Ontology Specification, *EUPASS project report*, 38 p.
- Mehrabi, M.G., Ulsoy, A.G. & Koren, Y. (2000). Reconfigurable manufacturing systems: key to future manufacturing. *Journal of Intelligent Manufacturing*, Vol. 11, No. 4, pp. 403-419, ISSN 0956-5515.
- Nylund, H., Salminen, K. & Andersson, P. (2008). Digital Virtual Holons - An Approach to Digital Manufacturing Systems, In: *Manufacturing Systems and Technologies for the New Frontier*, M. Mitsuishi et al. (Eds.), pp. 103-106, Springer, ISBN 978-1-84800-266-1.
- Salminen, K., Nylund, H. & Andersson, P.H. (2009). Role based self-adaptation of a robot DiMS based on system intelligence approach, *Proceedings of the 19th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2009)*, pp. 964-970, ISBN 978-0-9562303-2-4, July 6-8, 2009. Middlesbrough, UK.
- Siltala, N., Hoffmann, A., Gengenbach, U., Traivani, E. & Vollmer, H. (2008). EUPASS std-0004: Eplacement Specification and Guideline. *EUPASS project report*, 61 p.
- Smale, D. & Ratchev (2009). A Capability Model and Taxonomy for Multiple Assembly System Reconfigurations, *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, pp. 1923-1928, ISBN 9783902661432, 3-5, June, 2009, Moscow, Russia.
- Tolio, T. & Valente, A. (2006). An Approach to Design the Flexibility Degree in Flexible Manufacturing Systems, *Proceedings of the 16th Int. Conference on Flexible Automation and Intelligent Manufacturing (FAIM2006)*, pp. 1229-1236, 26 - 28, June, Limerick, Ireland.

- Ueda, K., Markus, a, Monostori, L., Kals, H. & Arai, T. (2001). Emergent Synthesis Methodologies for Manufacturing. *CIRP Annals - Manufacturing Technology*, Vol. 50, No. 2, pp.535-551, ISSN 0007-8506.
- Vichare, P., Nassehi, A., Kumar, S. & Newman, S.T. (2009). A Unified Manufacturing Resource Model for representing CNC machining systems. *Robotics and Computer-Integrated Manufacturing*, Vol. 25, No. 6, pp. 999-1007, ISSN 0736-5845.
- Wiendahl, H. et al. (2007). Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals - Manufacturing Technology*, Vol. 56, No. 2, pp. 783-809, ISSN 0007-8506.
- Wiendahl, H.P. & Heger, C.L. (2004). Justifying Changeability. A Methodical Approach to Achieving Cost Effectiveness. *Journal for Manufacturing Science and Production*, Vol. 6, No. 1-2, pp. 33-40, ISSN 0793-6648.

IntechOpen



Manufacturing System

Edited by Dr. Faieza Abdul Aziz

ISBN 978-953-51-0530-5

Hard cover, 448 pages

Publisher InTech

Published online 16, May, 2012

Published in print edition May, 2012

This book attempts to bring together selected recent advances, tools, application and new ideas in manufacturing systems. Manufacturing system comprise of equipment, products, people, information, control and support functions for the competitive development to satisfy market needs. It provides a comprehensive collection of papers on the latest fundamental and applied industrial research. The book will be of great interest to those involved in manufacturing engineering, systems and management and those involved in manufacturing research.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Eeva Järvenpää, Pasi Luostarinen, Minna Lanz and Reijo Tuokko (2012). Adaptation of Manufacturing Systems in Dynamic Environment Based on Capability Description Method, Manufacturing System, Dr. Faieza Abdul Aziz (Ed.), ISBN: 978-953-51-0530-5, InTech, Available from:

<http://www.intechopen.com/books/manufacturing-system/adaptation-of-manufacturing-systems-in-dynamic-environment-based-on-capability-description-method>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen