

Formal Resource and Capability Descriptions Supporting Rapid Reconfiguration of Assembly Systems*

Eeva Järvenpää, Niko Siltala, and Minna Lanz

Abstract— Today’s production environment is characterised by frequent changes in terms of high product variation, small batch sizes, high demand fluctuation as well as random unexpected disturbances on the factory floor. Production systems need to be rapidly reconfigurable and adaptable to these changing requirements. ReCaM project targets to develop a set of integrated tools for rapid and autonomous reconfiguration of production systems. Such tools need to be supported by formal information models describing the product requirements, as well as resource characteristics and functionalities. This paper concentrates on introducing the formal resource and capability models, which are used and further enriched to support ReCaM targets. Also examples of how these models can be applied to support rapid reconfiguration will be given.

I. INTRODUCTION

The requirements on production systems are continuously being shifted towards higher flexibility and adaptability. Increasing volatility in the global and local economies, shortening innovation and product life cycles, as well as a tremendously increasing number of variants, call for production systems, which comply with these changing demands. There is a need for rapidly responding production systems that can timely adjust to the required changes in processing functions, production capacity, and the dispatching of the orders. Responsiveness is fast becoming a new strategic goal for manufacturing enterprises alongside with quality and costs [1]. System reconfiguration is required on three levels: physical (changing the layout of the system, adding or removing machines or machine elements); logical (changing the process sequence, re-routing or re-scheduling production); and parametric (changing the adjustable machine parameters).

Despite the high efforts towards reconfigurable production systems e.g. by multitude of different EU-funded projects, and several standardization activities focusing on unification of mechanical as well as communication and control interfaces, reconfiguration of assembly systems is still rare in real factories. The usual business today, when the product model changes, is to scrap the existing resources and build a new assembly system from a scratch. This is due to high engineering, integration and programming efforts and skills needed to re-configure the existing system, as well as uncertainties related to the needed effort. One of the reasons for infeasibility of reconfiguration is the lack of sufficient information and documentation about the capabilities of the current system, its lifecycle, and usage history [2]. Therefore, standardisation of hardware and software interfaces is not, in itself, enough to enable the rapid and efficient reconfiguration.

Furthermore, there still remains work to be done in order to make these mandatory enabler interfaces into practice. Moreover, efficient methodologies, tools and information models are needed to support planners and engineers in the reconfiguration planning, integration and execution process, and also to allow logical and parametric reconfiguration to take place autonomously while the system is running.

The European Commission funded project ReCaM (Rapid Reconfiguration of Flexible Production Systems through Capability-based Adaptation, Autoconfiguration and Integrated Tools for Production Planning) [3], started in November 2015, aims to find solutions for the above mentioned issues. It targets to develop a set of integrated tools for rapid and autonomous reconfiguration of production systems, integrated with the existing production planning and scheduling tools (MES). The ReCaM approach is based on intelligent plug-and-produce capable self-describing Mechatronic Objects (MOs), which are able to auto-program and self-adjust to the required task by utilizing parametric capabilities. ReCaM approach will rely on an unified functional description of modules, providing a foundation for rapid creation of new system configurations.

This paper concentrates on describing the formal resource and capability models, which are utilized and further enriched to support ReCaM targets. Modifications implemented and designed to the models, to improve their performance, are highlighted. Also the roles of these two, previously independent, models in ReCaM context is discussed. Furthermore, examples of how these models can be applied to support rapid reconfiguration of production systems will be given.

II. EXISTING APPROACHES TO MODEL RESOURCES AND THEIR CAPABILITIES

The aim of bringing automation to the system design, re-configuration and order dispatching, requires a formal, structured representation of the product requirements as well as resource capabilities, properties and constraints. Recently, manufacturing resource modelling has been addressed by several researchers using different methods, having different purposes, from different viewpoints and at different levels of detail.

From the beginning of the millennium, there has been an increasing interest in manufacturing domain on using emerging technologies such as ontologies, semantics and semantic web, to support the collaboration, interoperability

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

All authors are with the Department of Mechanical Engineering and Industrial Systems, Tampere University of Technology, Tampere, Finland

(E. Järvenpää, phone +358-40-8490869, email: eeva.jarvenpaa@tut.fi; N. Siltala email: niko.siltala@tut.fi; M. Lanz email: minna.lanz@tut.fi).

and adaptation needs. In the context of distributed intelligent systems, such as agent-based or holonic systems, ontologies play a key role as they provide a shared, machine-understandable vocabulary for information exchange among dispersed agents. [4]

FP6 project PabadisPromise [5] resulted a manufacturing ontology (P2 ontology) and reference architecture focusing on factory floor control. Borgo and Leit [6] developed ADACOR ontology for distributed holon-based manufacturing focusing on processes and system interaction descriptions. It consists of ontological classification of ADACOR concepts according to DOLCE foundational ontology. During the FP6 EUPASS-project, an ontology for modelling evolvable, modular, ultra-precision assembly systems was developed [7]. Kitamura et al. [8] presented ontological definition of an assembly device capabilities based on the function-behaviour-structure (FBS) framework.

An ontology-based capability management approach for multi-agent-based manufacturing systems was developed by Timm et al. [9]. In the SIARAS project an intelligent system, called the skill server, was built to support automatic and semi-automatic reconfiguration of production processes [10][11]. Barata et al. [12] presented a multi-agent-based control architecture for a shop floor system (CoBaSa) which supports fast re-engineering and plug and play capabilities based on skill descriptions. Frei [13] applied the CoBaSa in Self-Organizing Evolvable Assembly Systems (SO-EAS). Also Obitko et al. [14] proposed ontology for agent-based manufacturing systems. Terkaj et al. [15] developed an ontological Virtual Factory Data Model, which acts as a shared meta-language providing a common definition of the data that are shared among different software tools along factory process lifecycle. In SkillPro-project, the classical product-process-resource concept was extended with the concept of skills. AutomationML-based format was used to store and communicate the skill descriptions to facilitate autonomous setup and execution of production tasks [16].

Manufacturing-as-a-Service paradigm has been in the interest of many researchers, who have produced different approaches to formally describe the service requests and offerings. Manufacturing Service Description Language (MSDL) was developed as a formal domain ontology for representation of capabilities of manufacturing services, focusing on mechanical machining services [17]. Later on, it has been extended for other applications, such as metal casting [4]. Shin et al. [18] enriched the MSDL further to comply better with the requirements of Manufacturing Service Capability (MSC) models. Hu et al. [19] developed an ontology-based digital description of resource services for grid manufacturing. In ManuCloud-project an XML-based manufacturing service description was developed to enable Manufacturing-as-a-Service operation principle in production network [20].

Most of the available approaches are domain-specific and offer only partial solutions for very specific applications, missing a comprehensive view. Also the ontologies or other data models are not publicly available, making their re-use practically impossible. The previous research attempts to describe manufacturing capabilities are limited in that they either don't consider the combined capabilities of multiple co-

operating resources, or they do not incorporate parameter information into the capability description. Furthermore, most of the presented approaches rely on static resource descriptions lacking the lifecycle aspect. In the context of production system re-configuration, information about the actual capability is needed instead of catalogue information.

III. INTRODUCTION TO THE FORMAL RESOURCE AND CAPABILITY DESCRIPTIONS

ReCaM-project utilizes, combines and enriches the existing formal capability model and resource description approach developed by the authors in previous projects. These independent models have been originally described in [2][21][22], and will be discussed in the following subsections. Furthermore, the modifications implemented and planned to these models is highlighted.

A. Capability model

Capability model is a data model for describing capabilities of resources. It includes the high level conceptual model for defining the concepts and their relations, as well as the formal ontology defining the actual capabilities, their relationships and detailed structure of the model. Capabilities are characterized by name and parameters. The capability concept name indicates the natural name of the capability, such as “moving”, “drilling”, “screwing”, and “grasping”. Capability parameters describe the characteristics of a capability, e.g. the “moving” capability is characterized by “velocity” and “acceleration” parameters, among others. The capability parameters help to distinguish between different resources which have similar capabilities. In other words, the concept name of the capability indicates the operational functionality of the resource, whereas the capability parameters determine the range and constraints of the capability.

Capability model (Fig. 1) divides the capabilities into simple and combined capabilities. Combined capabilities are upper level capabilities, which can be divided by functional decomposition into simple, lower level capabilities (*part_of* hierarchy). Combined capabilities are combinations of two or more (simple or combined) capabilities. In the model, the simple and combined capabilities are linked by capability associations. There are two types of capability associations, namely inputs and outputs. The simple capabilities provide output associations while the combined capabilities require input associations.

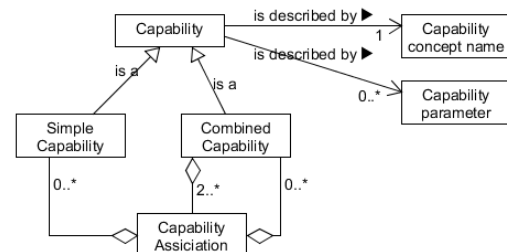


Figure 1. Concepts of the capability model.

The capabilities, modelled as classes in the ontology, form the capability catalogue, which consists of the pool of capabilities that may exist in a production system, including

their parameters. The simple capabilities can be assigned to resources through the resource description. When these generic capabilities are assigned to the resources, the capability parameters are filled with the resource-specific values.

Based on the defined capability associations, the resource combinations contributing to a certain combined capability can be identified and queried. Fig. 2 Figure 2. shows an example on how the capability associations are used in the ontology to connect the simple capabilities into combined capabilities. For instance, in order to transport an item the system needs to be able to move within some workspace and to hold the item. Therefore, both the “movingAssociation” and “holdingAssociation” have to be satisfied. Several different capabilities may provide output for a certain capability association. For instance, “holdingAssociation” can be satisfied either by grasping (e.g. gripper) or holding by gravity (e.g. conveyor belt). Similarly the same input association may be required by multiple different combined capabilities. As an example “spinningTool” capability is part of both “screwing” and “drilling” combined capabilities.

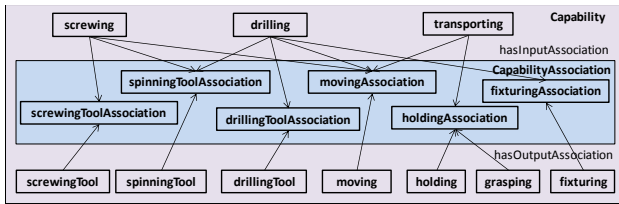


Figure 2. Example capabilities and capability associations between simple and combined capabilities.

When two or more resources are combined (e.g. robot + gripper), the associations between simple and combined capabilities allow the combined capabilities to emerge on the capability concept name level. Combined capability rules are needed to reason out the parameters of the combined capabilities (e.g. what is the payload of robot & gripper combination).

The capability model was previously a part of bigger ontology, called CoreOntology [2], which contains classes for modelling products, processes, resources and resource combinations. The ontology includes a taxonomy, which categorizes the capabilities in a hierarchical structure (e.g. “material removing” is a parent for “milling”, “drilling”, etc.) The taxonomy can be used to enable mapping between product requirements and resource capabilities at different levels of detail and allow subsumption-based reasoning about the capabilities. Both the product requirements and capability instances can refer to the capability taxonomy, which makes the matching possible on capability concept name level. Capability matching rules are used to make the match between product requirements and resource capabilities on parameter level.

B. Resource description concept

Resource description concept is a comprehensive XML-based digital representation of a technical entity. It integrates together information of a production resource related to geometrical, mechanical, functional, communication, and control aspects. It allows giving a description of resources’

functionality, interfaces to other resources, parameters related to business, environment and technical characteristics, as well as lifecycle related information. Resource description concept is a roof term and encapsulates detailed parts of descriptions and their inter relations, namely Abstract resource description (ARD), Resource description (RD), and Resource instance description (RID) (Fig. 3).

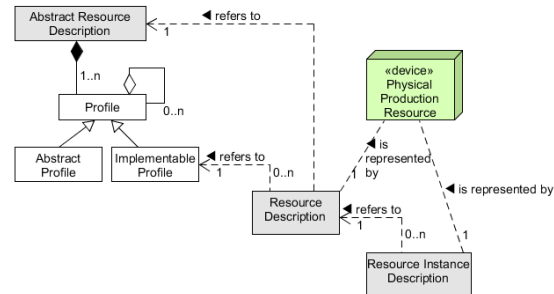


Figure 3. Resource description concept and relations between different descriptions.

Abstract Resource Description (ARD) is an abstraction and a reference model for production resources. It forms an abstract digital specification and generalisation for a collection of similar kind of production resources. In other words, ARD is a generalisation, which can be specialised as a physical production resource. ARD is composed of one or more Profiles, and it cannot be directly instantiated as a physical resource. Its purpose is to provide harmonisation over Resource Descriptions and its content is controlled by a harmonisation group(s).

Profile is an integral and inseparable part of ARD and cannot exist alone outside of it. Profile defines a reusable construction block of definitions, a structure which is used to specify the detailed section of the ARD. It includes information related to interfaces, capabilities, properties, and other features that are composing the generalisation of a set of production resources. A Profile can be built from N other Profiles with concepts of inheritance or referencing.

Resource Description (RD) is a digital representation of a real, physical production resource. It describes details associated to a specific type of HW resource used for production as a part of a production system. The description is jointly shared by same kind of resources i.e. resources having the same vendor, model, type, and version. It contains a reference to the ARD and Profile of which this resource claims to implement. RD represents the catalogue information of the resources.

Condition and capabilities of the resources evolve during their individual lifecycles and usages. Resource Instance Description (RID) is a digital representation of an individual physical instance of a resource. It carries the resources’ current state and historical data events – it is an accumulating information storage. It appends the RD with information that cannot be generalised over all instances of the same resource type, but is specific to a specific instance only. For instance, if the capability or lifecycle parameters (such as MTBF, Mean Time Between Failure) is changed during the resource lifecycle, RID will contain the updated information. The RID

should travel all the time with the physical production resource. Fig. 4 provides a detailed view of the three different descriptions, while Table 1 provides a concrete meaning of each description with example instances.

TABLE I. PURPOSE AND EXAMPLES OF DIFFERENT DESCRIPTIONS.

Term	Description and example	
	Description	Example
ARD	Represents specific technologies such as grippers, axis-systems or feeders. Each one of those has their own ARD, which collects all associated Profiles together.	ARD for grippers
Profile	Provides generalised specification of a specific kind of entity.	2-finger gripper
RD	RD turns the focus to the module provider (e.g. vendor V_A). It provides a detailed description of the specific production resource. This description respects the definitions made in Profile and ARD.	2-finger gripper from vendor V_A with type or model number T1.
RID	When vendor V_A produces physical entity of such 2-finger gripper of type T1, they give it a serial number. At the same time, they will create a RID and connect it to this specific piece of HW.	2-finger gripper from vendor V_A with type or model number T1 and serial number SN123

C. Adaptations to the models for ReCaM purposes

These two above introduced models have originally been developed completely independently from each other in different projects and with different objectives. In ReCaM-project the best parts of both models are utilized, and they are harmonized and coupled more tightly together. In ReCaM the capability ontology is used to model the capabilities, their parameters and associations between the combined and simple capabilities. In order to enable more powerful reasoning possibilities with the OWL-model, the structure of the original capability ontology, presented in [2], has been modified. For instance, in the new approach the capability concept names are implemented as classes instead of instances. This modelling method allows the capability parameters to be linked to the

specific capability classes as property restrictions. SWRL (Semantic Web Rule Language) rules are implemented directly to the ontology file to infer the combined capability parameters for the resource combinations based on the simple capabilities of the resources involved in the combinations. SPARQL queries are used to query information related to the capabilities and their parameters. Furthermore, in ReCaM the ontology has been distributed into three parts: capability model, product model and resource model. Both product model and resource model import the capability model to describe the product requirements and resource offerings.

The XML-based resource description, on the other hand, is responsible for handling other information related to resources, including interface specifications as well as business and lifecycle information. The resource description encapsulates the capability description of the resource it describes (as shown in Fig. 4). A new resource description editor will be developed during ReCaM for resource providers to ease up the filling of the resource related information. This editor will utilize the capability model (OWL) to describe the capability related aspects of the resource via referencing, while the other aspects will be directly captured in the XML-based resource description. After the resource description has been prepared, it is published in a resource catalogue and the capability related information can be read back to the ontology knowledge base. The reasoning related to capability matching is done in this knowledge base, utilizing SPARQL queries and SWRL rules. Thus, any non-capability related information saved into the resource description and needed for such matching, is mapped to the OWL format. An example of such information is weight of a gripper, which is needed in order to calculate the payload of a “robot & gripper” combination.

Both of the models have originally been developed to support mainly the design and reconfiguration planning phase of a production system. In ReCaM they are extended to support also the auto-programming and execution phase. Therefore, a new concept called executable capability has been added to the resource description (see Fig. 4). Executable capability is used for controlling the actual execution of the

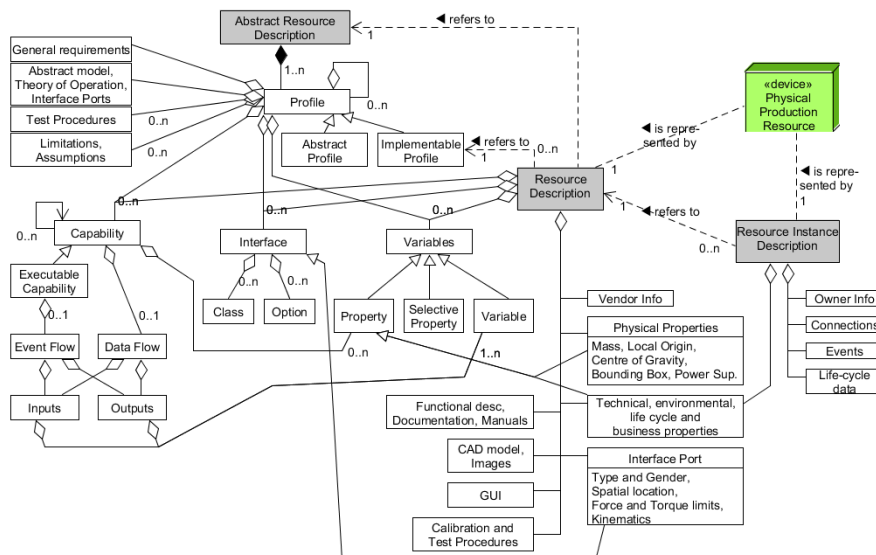


Figure 4. Detailed view to the Resource description concept, which now encapsulates the capability and executable capability descriptions.

capabilities existing on the resources, i.e. configuring the process parameters and triggering the execution of the capabilities. It differs from simple and combined capability descriptions by having input and output events, which triggers the activities and data inputs and outputs e.g. for setting up the process parameter values. The executable capabilities are part of the production recipe, which should be automatically generated and parameterized based on the product's process plan and available resources. Thus, the executable capabilities need to correspond to the actions that the resources make to complete the task goals (e.g. *moveTo*, *openFingers*, *closeFingers*).

IV. EXAMPLES OF THE MODELS' USAGE IN ReCAM

In order to speed up the reconfiguration of the existing production systems to new requirements, support for reconfiguration planning, auto-configuration and auto-programming is needed. This can be promoted e.g. by developing methods and tools which automate the processes of: 1) Evaluating the existing system against the new requirements; 2) Identifying any missing capabilities from the existing system; 3) Searching from resource libraries/catalogues for resources which satisfy the missing requirements; 4) Evaluating the interface compatibility of the resources; 5) Generating alternative system configurations; 6) Selecting the best configuration based on certain user defined criteria (e.g. smallest system change, maximising speed, or minimising energy consumption) while simultaneously complying with the requirements set by the production plan; 7) Generation of the recipe for the product's manufacturing/assembly process; 8) Supporting the plug-and-play capabilities of the resources and controllers; 9) Orchestrating the activities of the resources on the line according to the recipe; 10) Automatic deployment and commissioning of recipes and orchestration results.

The presented capability model and resource description provide support for all the above mentioned activities. In the following sections, few more detailed examples will be given on how the presented models support the ReCaM targets. The technical implementation of these examples is part of ongoing and future work of the project.

A. Common model for resource descriptions

The Resource Description (RD) is used to create a pool of resource descriptions, i.e. a Resource Catalogue. This catalogue may be a global library, where vendors can publish their resource offerings. On the other hand, the manufacturers' may use this library to search for suitable resources for their manufacturing requirements. Manufacturing companies may also have their in-house resource libraries, consisting of Resource Instance Descriptions (RIDs), which content is updated during the lifecycle of the resources.

The resource description and capability model provide a common terminology and semantics for different system component vendors to describe their system component capabilities and other characteristics in a digital form. This lays foundation for heterogeneous, multi-vendor systems with easy integration and ensures that there won't be ambiguity caused by different naming or structuring conventions between different vendors. Different SW-tools can retrieve

resource information from the catalogues and use it as digital representation of the resource.

In ReCaM-approach the system components, i.e. Mechatronic Objects, are intelligent modules which provide self-description. Thus the RID is carried by each individual MO. This description will support both offline and online planning, and also agent-based flow control.

B. Capability matching

The presented capability model provides basis for capability-based matching between products and resources. The product requirements are described by referring to the same capability model that is used to describe the resource capabilities. This allows rapid match-making between the requirements and capabilities, and fast identification of suitable candidate resources. In reconfiguration context the existing system can be compared against the new product requirements and the missing capabilities can be identified. New resources and resource combinations matching to the requirements can be searched from the research libraries, and the found matches can be suggested to the reconfiguration planning tool. Due to the ability to model the combined capabilities of combined resources (e.g. machine and its tooling), also small reconfiguration scenarios, such as tool changes, can be suggested.

The capability-based matching follows the principles of Service Oriented Architecture (SOA). In SOA, the services need to be described to the other parties in the system in a way, which allows the matching of the provider's offerings and the requestor's needs. This entails the services being described through their functional (what can they do?), behavioral (how is the functionality achieved?) and non-functional (constraints of the previous two) aspects. Following the approach of Kitamura et al. [8], the function represents the intention of the designer, i.e. what he or she wants to achieve. The behavior represents different manufacturing methods (capabilities) that can be used to achieve the required function. For example, the required function may be "joining", which can be achieved by several different behaviors, such as "welding", "gluing", "screwing", "riveting" and so on. The resource description and capability model facilitate the description of the behavior, whereas the capability taxonomy allows the capabilities to be linked to the related function. The non-functional aspects are handled through the capability parameters.

In SOA-based matching of the products and resources, the product is seen as a service requestor, whereas the order (including the product requirements) is seen as a service request. Resources are service providers, which provide the services through their capabilities. The role of the service broker is taken by the capability matching algorithms, which use the product requirements description, resource capability descriptions and the capability model defined by the ontologies, to make the match between requests and offerings.

The request may be targeted to either to a certain function, or if the process plan has already been defined, to certain behaviour (capability). This means that the request can be targeted at different levels in the capability taxonomy. If the request is targeted at functions, then all the resources having the capabilities which can implement that function will be offered. In ReCaM, the latter approach will be taken, targeting

the requirements to certain capability. This will make the matching more straightforward. On the other hand, targeting the requests on functions may be especially useful in product design and process planning phases, when possible available capabilities may be used as input for the design.

The capability-matching algorithm takes one capability requirement at a time, and matches it with the existing capabilities. If suitable capabilities are not found from the existing system, search can be targeted to the resource library, first in-house and finally global. The end result of the matching will be a list of resources matching to each capability requirement. This information will be provided to the reconfiguration planning tool, which will make the decision about resource selection and system configuration based e.g. the availability and other criteria.

Combined with layout description the capability model and associated capability matching rules will provide necessary information for the logical adaptation (e.g. routing). For parametric adaptation the capability description defines the ranges for certain parameters (e.g. spinning speed). With the executable capability extension, also the auto-programming and execution phases will be supported. The executable capabilities can be parameterized on real time based on the product requirements by inputting the wanted parameter value, while the executable capability is triggered.

V. CONCLUSION

This paper presented an ontological capability model and XML-based resource description concept for describing manufacturing resources. The evolution of these models to improve their performance for ReCaM project targets was highlighted. Examples of their use in rapid reconfiguration context were given. Due to the formal specification, independence from implementation and platforms, generality and re-usability, both information models facilitate information sharing and interoperability between various design and planning tools used during the product, process, and system design and reconfiguration, as well as production planning and manufacturing execution phases.

The developed knowledge representation and associated reasoning modules facilitate automatic filtering of resource information and allow suitable candidate solutions to be found from vast amount of input information. This may be especially useful during the initial system design, when the global resource libraries are used to search for suitable resources. Furthermore, e.g. in large factories with thousands of resources this automatic filtering can provide remarkable savings in time used for system design, reconfiguration, production planning and order dispatching. The presented models and approaches ease and speed up the reconfiguration planning, and facilitate also the reactive adaptation. Currently the presented approaches have only been demonstrated in an academic research environment. During ReCaM project, the models will be utilized in real industrial contexts, and will be further developed based on the feedback received from the industry.

REFERENCES

- [1] Y. Koren, and M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of Manufacturing Systems*, 29(4), 2010, pp. 130-141.
- [2] E. Järvenpää, *Capability-based Adaptation of Production Systems in a Changing Environment*, PhD dissertation, Tampere University of Technology, 2012, 190 p.
- [3] ReCaM consortium, ReCaM project web page, <http://www.recam-project.eu>.
- [4] F. Ameri, C. Urbanovsky, and C. McArthur, "A Systematic Approach to Developing Ontologies for Manufacturing Service Modeling," *Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing*, 2012, 14 p.
- [5] Pabadis'Promise, *D3.1 Development of manufacturing ontology*, Project deliverable, The PABADIS'PROMISE FP6 project consortium, 2003.
- [6] S. Borgo and P. Leit, "Foundations for a core ontology of manufacturing," *Integrated Series in Information Systems*, vol 14, 2007.
- [7] N. Lohse, *Towards an ontology framework for the integrated design of modular assembly systems*, PhD dissertation. University of Nottingham, 2006, p. 245.
- [8] Y. Kitamura, Y. Koji, and R. Mizoguchi, "An ontological model of device function: industrial deployment and lessons learned," *Applied Ontology*, 1(3-3), 2006, pp. 237-262.
- [9] I.J. Timm, T. Scholz, and O. Herzog, "Capability-based emerging organization of autonomous agents for flexible production control," *Advanced Engineering Informatics*, 20(3), 2006, pp. 247-259.
- [10] J. Malec, A. Nilsson, K. Nilsson, and S. Nowaczyk, "Knowledge-based reconfiguration of automation systems," *3rd Annual IEEE Conference on Automation Science and Engineering*, 2007, pp. 170-175.
- [11] M. Bengel, "Modelling objects for skill-based reconfigurable machines," *In Innovative production machines and systems*, 3rd I*PROMS Virtual International Conference 2007, p. 13.
- [12] J. Barata, L. Camarinhamatos, and G. Candido, "A multiagent-based control system applied to an educational shop floor," *Robotics and Computer-Integrated Manufacturing*, 24(5), 2008, pp.597-605.
- [13] R. Frei, G. Di Marzo Serugendo, N. Pereira, J. Belo, and J. Barata, "Implementing self-organisation and self-management in evolvable assembly systems," *IEEE International Symposium on Industrial Electronics*, 2010, pp. 3527-3532.
- [14] M. Obitko, P. Vrba, V. Marik, M. Radakovic, and P. Kadera, "Applications of semantics in agent-based manufacturing systems," *Informatica*, 34, 2010, pp. 315-330.
- [15] W. Terkaj, and M. Urgo, "Virtual Factory Data Model to support Performance Evaluation of Production Systems," *CEUR Workshop Proceedings*, Vol. 886, 2012, pp. 44-58.
- [16] J. Pfrommer, D. Stogl, K. Aleksandrov, V. Schubert, and B. Hein, "Modelling and Orchestration of Service-based Manufacturing Systems via Skills," *Proceedings of the IEEE Emerging Technology and Factory Automation*, 2014.
- [17] F. Ameri and D. Dutta, "An Upper Ontology for Manufacturing Service Description," *ASME Conference Proceedings*, 2006, pp. 651-661.
- [18] J. Shin, B. Kulvatunyou, Y. Lee, et al., "Concept Analysis to Enrich Manufacturing Service Capability Models," *Procedia Computer Science*, Vol 16, 2013, pp. 648-657.
- [19] Y. Hu, F. Tao, D. Zhao, and S. Zhou, "Manufacturing grid resource and resource service digital description," *The International Journal of Advanced Manufacturing Technology*, Vol. 44, Iss. 9-10, 2009, pp. 1024-1035.
- [20] U. Rauschecker, and M. Stöhr, "Using Manufacturing Service Descriptions for flexible Integration of Production Facilities to Manufacturing Clouds," *Proceedings of the 18th International Conference on Engineering Technology and Innovation*, 2012.
- [21] E. Järvenpää, P. Luostarinen, M. Lanz, and R. Tuokko, "Presenting capabilities of resources and resource combinations to support production system adaptation," *IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 2011 p. 6.
- [22] N. Siltala, and R. Tuokko, "Emplacement and Blue Print-Electronic Module Description Supporting Evolvable Assembly Systems Design, Deployment and Execution," *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*, 2010, pp. 773-788.