TAMPERE UNIVERSITY OF TECHNOLOGY

Formal resource and capability models supporting re-use of manufacturing resources

Citation
Järvenpää, E., Lanz, M., & Siltala, N. (2018). Formal resource and capability models supporting re-use of manufacturing resources. Procedia Manufacturing, 19, 87-94. https://doi.org/10.1016/j.promfg.2018.01.013

Year
2018

Version
Publisher's PDF (version of record)

Link to publication
TUTCRIS Portal (http://www.tut.fi/tutcris)

Published in
Procedia Manufacturing

DOI
10.1016/j.promfg.2018.01.013

Take down policy
If you believe that this document breaches copyright, please contact cris.tau@tuni.fi, and we will remove access to the work immediately and investigate your claim.

Download date:20.09.2019

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 19 (2018) 87–94

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

6th International Conference on Through-life Engineering Services, TESConf 2017, 7-8 November 2017, Bremen, Germany

Formal Resource and Capability Models supporting Re-use of Manufacturing Resources

Eeva Järvenpää*, Minna Lanz, Niko Siltala

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

Abstract

In the field of manufacturing the responsiveness has become a new strategic goal for the enterprises alongside with quality and costs. Efficient responsiveness requires production reconfiguration ranging from layout to equipment. The production system capabilities originate from the tool and equipment level. While a resource is being used, its condition and capability may change. It is crucial to consider the resources' individual lifecycle, their actual capabilities and condition during the system design and reconfiguration. Thus, the lifecycle perspective in the capability management is of utmost importance. This paper presents the development of the Manufacturing Resource Capability Ontology (MaRCO), focusing on describing the functional capabilities of manufacturing resources. Special emphasis is placed on the lifecycle management aspect of the resource descriptions.

© 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 6th International Conference on Through-life Engineering Services.

Keywords: Capability model; Production system representation; Adaptive manufacturing; Ontology

1. Introduction

Responsiveness has become a new strategic goal for the manufacturing enterprises alongside with quality and costs [1]. The responsiveness is related to the need to reconfigure and adjust the production and corresponding production

* Corresponding author. Tel.: +358408490869
E-mail address: eeva.jarvenpaa@tut.fi

2351-9789 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 6th International Conference on Through-life Engineering Services.
10.1016/j.promfg.2018.01.013

system as efficiently as possible to the required changes in processing functions, production capacity, and the dispatching of the orders. Furthermore, the trend towards sustainability requires that the resources are being utilized and re-used in their maximum potential and maintained throughout their life-cycle. Thus, the maintenance function becomes an important factor in terms of improving the system availability, reliability and safety, as well as product quality [2][3].

Information that is needed for reconfiguration and re-use decisions comes from different fields of expertise such as product design, process planning, system design and integration, operation, condition monitoring and maintenance, as well as usage history and prior adaptation experiences. The management of such information in industry is not currently sufficient to generate reliable plans for re-configuration and re-use. The real capabilities of the resources are not known in every lifecycle phase, which causes difficulties in making maintenance, re-configuration and re-use decisions. In large manufacturing facilities, or complex research environments there can be a huge amount of individual assets to manage. Document-based management of this information is inefficient. What makes the management of this information difficult is its rapidly evolving nature. Another problem, discussed by multiple researches, is the poor interoperability between different information models created by different design systems. The information is presented via different models that are usually complementary, but sometimes redundant, sometimes incoherent and always heterogeneous [4] as the majority of these systems use their proprietary data structures and vaguely described semantics. The design knowledge remains locked inside the authoring system and a lot of data may be lost during format conversions [5][6]. Retrieving and utilizing information from multiple diverse sources puts high demands on semantic integration solutions [1][7][8]. Therefore, a formal and semantic way to model the resources and their lifecycle information is needed in order to convey this information to the planning and decision-making activities.

As engineering practices are becoming more and more distributed and decentralized, formal engineering ontologies are emerging as popular solutions for addressing the semantic interoperability issue in heterogeneous environments and bridging the gap between the legacy systems and organizational boundaries [7][8]. The European Commission funded project ReCaM (Rapid Reconfiguration of Flexible Production Systems through Capability-based Adaptation, Auto-configuration and Integrated Tools for Production Planning) aims to develop a set of integrated tools for rapid and autonomous reconfiguration of production systems. The approach relies on a unified functional description of resources, providing a foundation for rapid creation of new system configurations through capability-based matchmaking of product requirements and resource offerings. This paper presents the development of the Manufacturing Resource Capability Ontology (MaRCO), intended to support such matchmaking. Special focus is placed on the lifecycle management aspect of the resource descriptions.

2. Formal information models for describing resources

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's capabilities, properties and constraints. For the past two decades, 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 and adaptation needs. Table 1 shows the literature review on the developed product, process and system ontologies and other formal information models. These models describe a set of product and system characteristics relevant to that specific domain. The table classifies the main developments from product, system, production control and operational maintenance strategy aspects. The identified gap is the lack of detailed capability descriptions of constantly evolving production systems.

Table 1. Existing ontologies and other formal information models for different life-cycle phases of products and systems.

Category	Existing works
Product design	Closed-loop product lifecycle management [5]; Reference ontology to support product life-cycle management [9]; Machine design process and associated ontology for design and lifecycle information of a specific machine [10].
System descriptions	Ontology for modelling evolvable, modular, ultra-precision assembly systems [6][11]; Emplacement concept [12]; Ontological model for assembly device capabilities based on the function-behaviour-structure (FBS) framework [13]; Manufacturing Service Description Language (MSDL) for representing capabilities of

	manufacturing services, focusing on machining [14]; Manufacturing Service Capability (MSC) models [15]; Ontology-based digital description of resource services for grid manufacturing [16]; Skillpro product-process-resource concept extended with skills [17]; Virtual Factory Data Model [18].
Production control	Manufacturing ontology (P2 ontology) [19]; ADACOR ontology for distributed holon-based manufacturing [7]; An ontology-based capability management approach for multi-agent systems [20]; Automatic and semi-automatic reconfiguration of production processes based on skills [21]; Ontology for agent-based manufacturing systems [22]; Multi-agent-based control architecture for a shop floor system (CoBaSa) [23]; Self-Organizing Evolvable Assembly Systems (SO-EAS) [24].
Maintenance planning	Ontology-based framework for collaborative maintenance planning in collaborative maintenance environment [3]; Industrial MAintenance Management Ontology (IMAMO) [4]; Monitoring and maintenance platform and corresponding ontology [2].

The previous research to describe manufacturing and system capabilities are limited in that they either do not 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 in the field of system description and production control rely on static resource descriptions lacking the lifecycle aspect. The existing approaches describe the nominal capability of the resources rather than their actual, current capability. In the context of production system re-configuration, re-use, re-tooling and maintenance the information about the actual capabilities are needed instead of catalogue information.

3. Development of Manufacturing Resource Capability Ontology for continuously evolving systems

Ontology engineering methodology [25] was utilized during the development of the Manufacturing Resource Capability Ontology (MaRCO). This methodology deals with the process and methodological aspects of ontology engineering, i.e. by providing guidelines on how to develop ontologies. The methodology consists of five phases: 1) Feasibility study, 2) Kickoff, 3) Refinement, 4) Evaluation, 5) Application and Evolution. This section concentrates on the results of the kickoff and refinement phases. In kickoff, the requirements for the ontology were detailed and important concepts and their relations were analysed. The end result of this activity was the conceptual model for modelling the evolving resource capabilities. The results of the kickoff phase are discussed in sections 3.1. and 3.2. from two perspectives, namely capability matchmaking and resource lifecycle management. In the refinement phase, the ontology was formalized and represented in OWL language. Section 3.3 discusses the results of the refinement phase.

3.1. Capability matchmaking perspective

The initial purpose of the developed ontology was to support automatic matchmaking between product requirements and resource capabilities. The device combinations are matched to the product requirements automatically based on their availability and combined capabilities. Thus, the essential concepts and their mutual relations in the areas of Product, Process, Capability, Resource, and System were first identified. Fig. 1 gives a conceptual description of how the product, resource, and capability concepts are linked together in the matchmaking. For greater simplicity and readability, only the most relevant identified relations are shown. The term "Device" is used to indicate that the work concentrates only on machine and tool resources, and not on other resource types, such as human operators or raw materials.

During the analysis, it was identified that four distributed ontologies are needed to facilitate such capability matchmaking process. These ontologies have different users, and different usage phases during the product and production system lifecycle. Instead of having one large ontology, the distribution reduces the complexity of the model from the user's point of view. Process Taxonomy Model defines the hierarchical categorization of different manufacturing processes, e.g. milling is classified as machining and further as material removing process. Product Model is used to model the product characteristics and manufacturing requirements. Capability Model defines the capability names, parameters and associations between simple and combined capabilities. Resource Model defines the resources and systems composed of the resources. This paper focuses on the Resource Model which imports the Capability Model, thus referred to here as the Manufacturing Resource Capability Ontology (MaRCO).

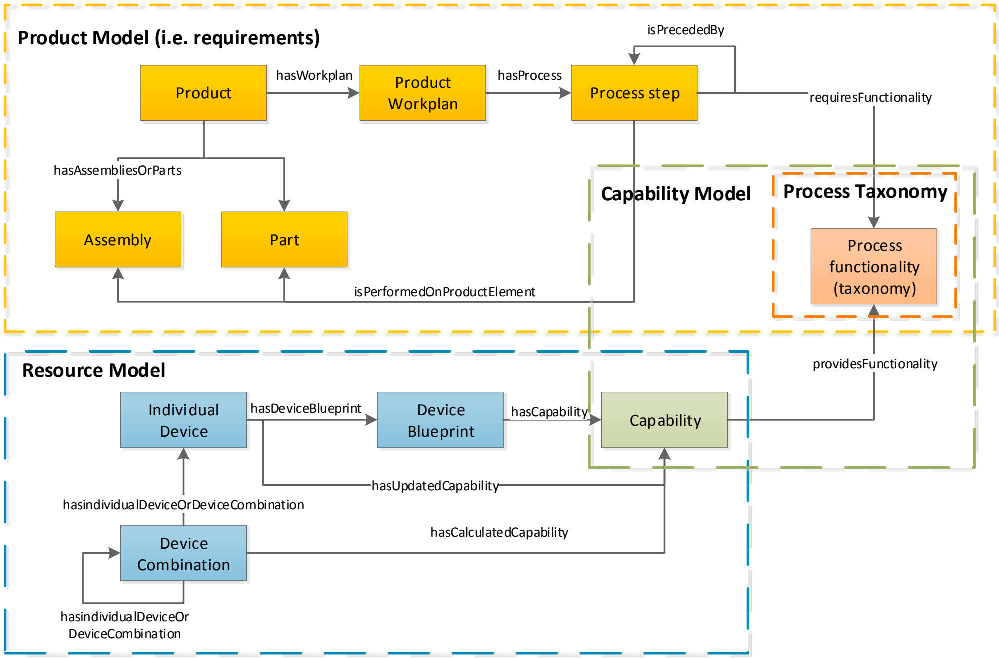

Fig. 1. Simplified model of the relations between product requirements, resources and capabilities.

The conceptual model of capabilities was initially presented in [26]. 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 "speed" and "acceleration" parameters, among others. The concept name of the capability indicates the operational functionality of the resource, whereas the capability parameters determine the range and constraints of that functionality. The Capability Model 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). The capabilities can be assigned to resources in the Resource Model ontology. When these generic capabilities are assigned to the resources, the capability parameters are filled with the resource-specific parameter values. Based on the defined capability associations between the simple and combined capabilities in the Capability Model ontology, the resource combinations contributing to a certain combined capability can be queried and identified on a concept name level. For more information about the capability-based adaptation, capability model and capability-matching, please refer to the original sources written by the authors [26] [27].

3.2. Resource life-cycle management perspective

The preliminary assumption is that the production environment is constantly changing, and the condition and capabilities of the resources evolve during their individual lifecycles and usages. The resource's condition, its remaining lifetime and consequently its capability will vary depending on the applications it has been used for, the environments it has been used in, the processes it has performed and the process parameters, which have been used. In order to support the system designers and maintenance planners, updated resource information needs to be made available for them.

Due to the lifecycle-based evolution of resource capabilities, there needs to be two descriptions of the resource: one describing certain type and model in its initial, i.e. nominal, condition, and other one describing the specific resource instance during its lifecycle. The latter description has to be dynamically updated over time. For this reason, the devices have two linked representations, in the ontology: device blue prints and individual devices.

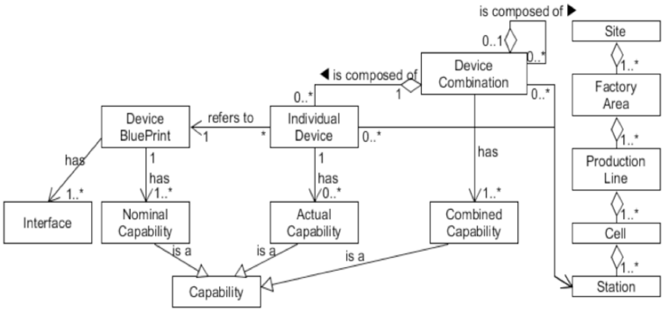

Fig. 2. Conceptual relations between resources and capabilities.

Fig. 2 illustrates the important concepts in the Resource Model. The device blue print describes the capabilities, interfaces and properties of one type and model of device, as given in the resource providers' catalogues. This represents the nominal capability of the device. The individual devices are presented as a separate concept, representing the actual capabilities of the particular, individual resource existing on the factory floor, and having a reference to the device blue print. If there is a number of similar machine individuals, they all will refer to the same blueprint device.

The individual devices have the actual capabilities, which are affected by the lifecycle of each individual device and updated according to measured or calibrated values from the factory floor. If there are no measured or updated values or other evidence of the actual capability of an individual device available, the capability of the individual device is assumed to be the nominal capability of the referenced item. This is the case, for example, when a new machine is taken into use at the first time. This approach allows the capability definitions to be reused, and furthermore, allows the accuracy of the description to be increased as more data becomes available.

The capability and/or its parameters may change e.g. due to maintenance operations, e.g. if some components have been replaced with new ones. For example, if the measured accuracy of the machine differs from the value defined in the nominal capability, for instance because of wear or maintenance, this updated value can be given in the actual capability definition. In some cases the breakdown can mean that only part of the capability is lost and the rest can be recovered. E.g. a Cartesian manipulator may lose its ability to move in y-direction, while it is still able to move in x- and z-directions. This means that just the degrees of freedom parameters of the "Moving" capability are changed and the resource can continue its operation with modified capability.

3.3. Main classes of the MaRCO model

In the refinement phase, the ontology was formalized and represented with the Web Ontology Language (OWL). OWL is a semantic mark-up language for publishing and sharing ontologies on the World Wide Web, and is developed as a vocabulary extension of RDF (Resource Description Framework) [28]. Protégé ontology editor [29] was used to construct the ontology. The main classes of the Capability Model are introduced in Table 2.

Table 2. The main classes of the Capability Model.

Class	Description
Capability	This is a parent class for the specific capability classes, which define the functionalities of resources. Sub-classes include all simple and combined capabilities currently modelled. Capability parameters are implemented as datatype and object properties, and are linked to the capabilities by property restrictions. Property restrictions are used to restrict the properties of instances belonging to a certain class. Most of the capability parameters are implemented as

	datatype properties, which can be used to save literal data values. Some parameters are implemented as object properties, which are used to model relationships between instances. These instances belong to CapabilityParameterAdditional class, discussed below. Example of capability definition can be seen in Fig. 3(a), which represents the class definition of "MillingCutterFunction" capability.
CapabilityAssociation	This class is meant for storing capability association instances, which are used to create links and hierarchy between simple and combined capabilities in the model. They are linked to the capabilities by hasOutputAssociation and hasInputAssociation object properties with hasValue property restriction. For example, for all instances belonging to the "Milling" class, the hasInputAssociation object property has values "millingToolAssociation", "spinningToolAssociation", "movingAssociation" and "fixturingAssociation".
CapabilityParameter Additional	This is a parent class for storing capability parameter groups, which are not feasible to be stored as individual datatype properties, but rather as instances linked through object properties. This approach is utilized e.g. for such capability parameters, which form a natural group, which relate to multiple different capabilities. Other example is parameter groups, which relate directly to a certain capability, but which properties depend on the nature of the capability. Examples of the sub-classes include: Workspace, MovementRange, ItemSize, ShapeDefinition and BasicResourceInformation.
ProcessTaxonomy (imported)	The process taxonomy defines hierarchical classification of different process functionalities. It is a pure taxonomy, without any properties. The taxonomy can be used to link the capabilities to upper level in the process hierarchy, e.g. "Milling" would be classified as a "Material removing" process. Capability Model imports the Process Taxonomy, and the capability classes are defined as the sub-classes of the relevant process taxonomy classes. The reasoning ability of OWL allows then to directly infer that each instance saved to a certain capability class is also an instance of the process taxonomy class, which was defined as the parent class of the specific capability.

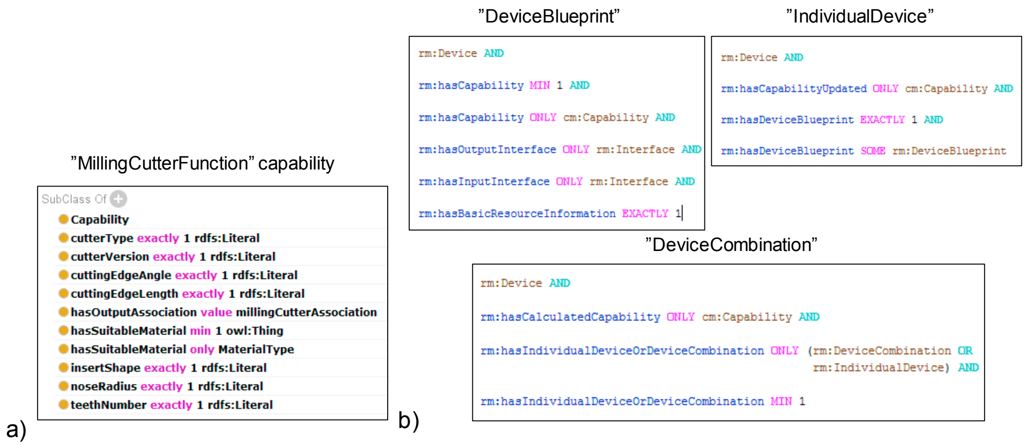

Fig. 3. (a) Property restrictions defined for "MillingCutterFunction" capability; (b) Class definitions of DeviceBlueprint, IndividualDevice and DeviceCombination.

The Resource Model ontology imports the above presented Capability Model. Together they form the Manufacturing Resource Capability Ontology (MaRCO). The formalization of the Resource Model follows quite closely to the semi-formal ontology definition presented in Fig. 2. In the following the most relevant classes in the Resource Model are shortly described. Moreover, Fig. 3(b) presents the class definitions for DeviceBlueprint, IndividualDevice and DeviceCombination. These classes are in the core of the developed modelling concept as they link the capabilities to the devices, both catalogue and actual existing device individuals, and allow modelling of device combinations for which combined capabilities can emerge. Table 3 introduces the most relevant classes in the Resource Model ontology.

Table 3. The main classes of the Resource Model ontology

Class	Description
Resource	This is a parent class for all Resource related classes. Direct sub-classes are Device, FactoryUnit, Human, RawMaterial and Software.

Device	This is a parent class for different Device related classes. It includes any machines, equipment and tools used to manufacture or assemble a product. Devices can be either catalogue devices (blueprints), actual individual devices or combinations of them.
DeviceBlueprint	This class contains the catalogue information of devices. This class has several subclasses for different categories of devices, such as FeedingDevice, GraspingDevice, MovingDevice and ModifyingDevice. The instances of DeviceBlueprint relate to specific Capability instances by hasCapability object property. These capability instances are filled with resource specific capability parameter values, making the capability description unique for each instance of the DeviceBlueprint class.
IndividualDevice	The instances of this class represent the actual individual devices existing on the factory floor. The individual devices have reference to the DeviceBlueprint through hasDeviceBlueprint object property. This class stores the lifecycle information of the devices and updates the capability properties. The updated capabilities are modelled through hasCapabilityUpdated object property.
DeviceCombination	This class includes instances, which represent combinations of multiple individual devices or device combinations. These instances refer to the instances of IndividualDevice or DeviceCombination class through hasIndividualDevicesOrDeviceCombinations object property. Furthermore, for the DeviceCombination instance, combined capability information can be saved through hasCalculatedCapability object property.

4. Discussion and Conclusions

This paper introduced the Manufacturing Resource Capability Ontology (MaRCO) model that allows creation of detailed capability descriptions of evolving resources used in the factory floor. The MaRCO model offers methods to formally describe and store not only the nominal capability of a certain resource type, but also the updated information of the resource instances through their use and individual lifecycle. Therefore, it provides the planners more insightful information to support their decisions on the reconfiguration, re-use and maintenance of the resources. The ontology is currently in evaluation and testing phase.

MaRCO is optimized for the capability matchmaking and consequently for the information that is needed for such matchmaking process. The fact that certain capabilities exist in the system, does not necessarily mean that those capabilities could be used. For instance, a machine may break down often, which means that its MTBF is short. This does not necessarily directly affect to the capability description of the machine, if the capability parameters are not affected by the machine condition which causes the breakdowns. However, if the resource is not available, none of its capabilities can be used. For example, lathe having “Turning” capability, cannot perform turning while not available, which means that the availability of the “Turning” capability is also zero. In case of multifunctional lathe, which has “Turning”, “Milling” and “Drilling” capabilities, none of these capabilities can be used while the machine is in maintenance or service. If the maintenance is successful, the capabilities become again available. These are important factors to take into account during the system design and especially during production planning and control. Currently, MaRCO doesn’t consider these aspects. In the future, we will use the XML-based Resource Description Concept [30] to save additional information, such as business and lifecycle parameters (e.g. MTBF, MTTR, operating costs) of the manufacturing resources. The production, reconfiguration and maintenance planners can then make use of this additional information after the automatic matchmaking system, relying on the MaRCO model, has detected the resources matching with the capability requirements.

Acknowledgements

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

References

[1] Y. Koren, M. Shpitalni, Design of reconfigurable manufacturing systems, J. of Manufacturing Systems, 29(4) (2010) 130-141.

[2] A. Chioreanu, S. Brad, C. Porumb, S. Porumb, E-maintenance ontology-based approach for heterogeneous distributed robotic production capabilities, International Journal of Computer Integrated Manufacturing, 28:2 (2015) 200-212.

[3] R. Genquan, Z. Yinwen, Z. Li, W. Jianmin, L. Ting, An Ontology-Based Framework for Collaborative Maintenance Planning. In: A. König, A. Dengel, K. Hinkelmann, K. Kise, R.J. Howlett, L.C Jain (eds), Knowledge-Based and Intelligent Information and Engineering Systems. KES 2011. Lecture Notes in Computer Science, vol 6881 (2011). Springer, Berlin, Heidelberg.

[4] M.H. Karray, B. Chebel-Morello, N. Zerhouni, A Formal Ontology for Industrial Maintenance, Terminology & Ontology: Theories and applications, TOTh Conference, (2011) Annecy, France. 1-20.

[5] A. Matsokis, D. Kiritsis, An ontology-based approach for Product Lifecycle Management. Computers in Industry, Vol. 61, Iss. 8 (2010) 787-797.

[6] N. Lohse, Towards an ontology framework for the integrated design of modular assembly systems. PhD thesis. University of Nottingham (2006) p. 245, ISBN 0387312765.

[7] S. Borgo, P. Leitão, Foundations for a core ontology of manufacturing, Integrated Series in Information Systems, 14 (2007).

[8] F. Ameri, C. Urbanovsky, 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.

[9] G. Bruno, D. Antonelli, A. Villa, A reference ontology to support product lifecycle management, 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14, Procedia CIRP 33 (2015) 41 – 46.

[10] J. Salonen, et al., An implementation of a semantic, web-based virtual machine laboratory prototyping environment, In: L. Aroyo, et al. (eds.), The Semantic Web ISWC 2011 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part II. LNCS vol. 7032 (2011) 221-236.

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

[12] N. Siltala, R. Tuokko, Emplacement and Blue Print–Electronic Module Description Supporting Evolvable Assembly Systems Design, Deployment and Execution. In: G.Q. Huang, K.L. Mak, P.G. Maropoulos (eds.), Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology (2010) 773–788.

[13] Y. Kitamura, Y. Koji, R. Mizoguchi, An ontological model of device function: industrial deployment and lessons learned. Applied Ontology, 1(3-3) (2006) 237–262.

[14] F. Ameri, D. Dutta, An Upper Ontology for Manufacturing Service Description, ASME Conference Proceedings, (2006) 651-661.

[15] J. Shin, B. Kulvatunyou, Y. Lee, et al., Concept Analysis to Enrich Manufacturing Service Capability Models," Procedia Computer Science, Vol 16 (2013) 648-657.

[16] Y. Hu, F. Tao, D. Zhao, S. Zhou, Manufacturing grid resource and resource service digital description, The International Journal of Advanced Manufacturing Technology, 44(9-10) (2009) 1024-1035.

[17] J. Pfrommer, D. Stogl, K. Aleksandrov, V. Schubert, B. Hein, Modelling and Orchestration of Service-based Manufacturing Systems via Skills, Proceedings of the IEEE Emerging Technology and Factory Automation (2014).

[18] W. Terkaj, M. Urgo, Virtual Factory Data Model to support Performance Evaluation of Production Systems,CEUR Workshop Proceedings, Vol. 886 (2012) 44-58.

[19] Pabadis'Promise, D3.1 Development of manufacturing ontology. Project deliverable, The PABADIS'PROMISE FP6 project consortium (2003).

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

[21] M. Bengel, Modelling objects for skill-based reconfigurable machines. In Innovative production machines and systems, 3rd I*PROMS Virtual International Conference (2007) p. 13.

[22] M. Obitko, P. Vrba, V. Marik, M. Radakovic, P. Kadera, Applications of semantics in agent-based manufacturing systems. Informatica, 34 (2010) 315-330.

[23] J. Barata, L. Camarinhamatos, G. Candido, A multiagent-based control system applied to an educational shop floor. Robotics and Computer-Integrated Manufacturing, 24(5) (2008) 597-605.

[24] R. Frei, G. Di Marzo Serugendo, N. Pereira, J. Belo, J. Barata, Implementing self-organisation and self-management in evolvable assembly systems. In 2010 IEEE International Symposium on Industrial Electronics. IEEE (2010) 3527-3532.

[25] Y. Sure, S. Staab, R. Studer, Ontology Engineering Methodology, In: S. Staab, R. Studer (Eds.), Handbook on Ontologies, 2nd edition (2009) 135-152.

[26] E. Järvenpää, Capability-based Adaptation of Production Systems in a Changing Environment. PhD Thesis, Tampere University of Technology (2012) 190 p. ISBN 978-952-15-2928-3. Available in: http://URN.fi/URN:ISBN:978-952-15-2940-5

[27] E. Järvenpää, N. Siltala, M. Lanz, Formal Resource and Capability Descriptions Supporting Rapid Reconfiguration of Assembly Systems, In Proceedings of the International Symposium on Assembly and Manufacturing, IEEE (2016) 120-125.

[28] W3C, OWL Web Ontology Language – Reference (2004) Available in: http://www.w3.org/TR/owl-ref/ [Accessed 1.11.2015].

[29] Protégé, "Protégé Ontology Editor web page", Available in: http://protege.stanford.edu/ [Accessed 1.11.2015].

[30] N. Siltala, E. Järvenpää, M. Lanz, Formal Information Model for Representing Production Resources, Advances in Production Management Systems: Initiatives for a Sustainable World - IFIP WG 5.7 International Conference, APMS 2016, Springer (2016) 53-60.