Cyber-physical robotics – automated analysis, programming and configuration of robot cells based on Cyber-Physical-Systems

Joachim Michniewicz, Gunther Reinhart

Abstract

This paper shows a Cyber-Physical-System based approach which allows the efficient utilization of the entire flexibility of reconfigurable, modular robot cells in assembly. Robots cells are automatically programmed, configured and optimized based on the solution independent virtual requirements of the individual product. A detailed virtual representation of the entire robot cell eliminates uncertainties regarding the feasibility of the assembly process in a defined robot cell during the design phase of the product with a CAD program. The necessary virtual representations of the robot cell components and the products to be manufactured are defined and the information interchange is explained.

© 2014 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

Peer-review under responsibility of the Organizing Committee of SysInt 2014.

Keywords: Cyber-Physical-Systems; reconfiguration and programming of robot systems; task description language; assembly; computer aided design

1. Introduction

Industrial robots have a great inherent flexibility due to their kinematical degrees of freedom and the versatility of manageable tools, sensors and other periphery devices. The effort needed to program and configure the entire robot system, for example at the introduction of a new or altered product, is high and is limiting the utilized flexibility.

* Corresponding author. Tel.: +49-89-289-15476; fax: +49-89-289-15555
E-mail address: Joachim.Michniewicz@iwb.tum.de
Therefore robots in the industrial environment are mostly used for repetitive, pre-defined tasks with little variance and adaptability [1].

This conflicts with the trends the manufacturing industry faces nowadays: growing uncertainty about product life-cycles, increasing product variance, globalization and shrinking lot sizes [2]. These demands have to be met with new approaches to program and configure robot systems, minimizing the required time and operator experience [3].

When a new product or a new variant is designed, there is no certainty about the feasibility of the manufacturing process on the existing installations. Descriptions of the manufacturing process are generated and analyzed manually. In a next step human robot programmers write the control code. If the product cannot be manufactured, either the product is altered or the robot system has to be changed. The communication between the operators and the designers lacks continuity due to the usage of no or incompatible types of software, resulting in a try-and-error proceeding. Time to market is increased and the planning reliability is reduced [4].

In order to facilitate the reconfiguration of robot cells, the Plug&Produce approach was introduced. By adding a storage device to the components containing its virtual description and corresponding drivers the effort for the reconfiguration of components from heterogeneous manufacturers could be reduced. The exchangeability of actuators and tools can be increased significantly [5, 6, 7].

In this paper the Plug&Produce approach is brought to the next level by adding a virtual representation of the product to the system, which contains its individual properties and requirements. The robot cell components and the product to be manufactured are defined as Cyber-Physical-Systems (CPS) [8, 9], allowing them to store data, to process data intelligently, to interact and to communicate with each other. The aim of Cyber-Physical-Robotics is the development of a method for an automated analysis, programming and configuration of every kind of robot cells. The necessary information is provided by solution independent data from the product. An economic automated production of all facility-adequate products and the utilization of the entire flexibility of robot cells are to be achieved.

In order to not only shift the effort of programming and configuring robot cells to the manual generation of the virtual representations of the products to be manufactured, its automated generation from CAD files and sensor data is researched and described. By connecting the virtual representation of the robot cell, which fully describes its skills, with the CAD program, the feasibility of the automated manufacturing process of the product can be analyzed without the need for real world testing.

To reach the before mentioned goals, a new system integrating partial solutions already existing in scientific literature has to be developed. Operations inside its necessary subsystems as well as the data exchange between them are explained and additionally new approaches, required to reach the technological vision, are presented in this paper. The paper focuses on the information exchanged and stored by various CPS in the robot cell and does not address data exchange technologies.

2. Vision

The presented approach can be seen as a necessary step to reach the vision of the future of industrial robots, which is explained in this chapter. A possible scenario could be: a product designer finishes the work on his project in his CAD software. The virtual representation of all product requirements towards the assembly process is generated from a CAD file. The designer sends the virtual description to a service provider for automated assembly. The service provider has a shop floor with a modular Cyber-Physical-Production-System (CPPS) with a variety of tools, sensors, mobile and fixed robots. All these devices have a virtual representation of their skills and properties. These are linked to a network and communicate with each other. Analysis of mutual interferences and interdependencies of the devices in the modular production system allow a precise determination of the available processes with the corresponding constrains. Processes required by the product can be automatically compared with the available processes. Information about product requirements exceeding abilities of the CPPS can be used to alter the product or be used by the service provider to add more appropriate devices to the CPPS. After determining the requirement fulfillment of the production process, the designer, being the client, can choose the optimization criterion (e.g. time, price, energy efficiency) for production planning. The designer can compare offers from various assembly service providers. The flexibility and modularity of the production systems allows the efficient production
of different products in various lot sizes. Due to the precise, fine grained description of the skills of the components, the production planning process can depict the whole flexibility of the CPPS on a previously unattainable level. After a service provider is chosen, programming and configuration is automatically derived from the virtual representation of the product.

3. Related work

In order to enable the product to communicate its requirements, a virtual description of those needs to be generated. The automated extraction of assembly sequences from virtual product data usually needs additional, application specific information. The files describing the product contain the geometrical information of every single part and the geometrical constrains between them. Assembly orders and movements can be determined from interrelations between the parts, their accessibility and collision-freedom in various assembly steps. A popular way to automatically determine assembly sequences is the description of interrelations between the parts with predefined features, which are optimized for the use case [10, 11, 12]. An alternative approach is knowledge based reasoning. Existing assembly plans of earlier products are modified to adapt to the requirements of a new product [13, 14, 15].

Commercial computer aided design systems do not offer the extraction of assembly sequences from CAD models. [16] analyze common CAD-tools and describe a promising approach for the automated generation of assembly sequences from unaltered files without the need of additional data. The presented assembly sequence generators do not create data, which can be easily used to program and configure a robot. [17, 18, 19] present systems combining geometrical information of each part with its interrelations to other parts in order to extract necessary assembly operations. Those are subdivided into sub-operations, executable on a predefined robot system.

As presented in the vision, the devices in the production system must have their individual virtual representation and be able to communicate in a network. This virtual representation can be used for a variety of functions. [20] introduce a Plug&Produce approach, in which the various devices automatically determine their position to each other. Complex assembly tasks are entered manually and are automatically decomposed into standardized primitive operations, which are allocated to the available assembly devices in real time. [21] present a service oriented system for describing reconfigurable manufacturing systems. These are modularized into physical subsystems, each of which has a digital description of its own skills and functions, based on Petri nets. The single Petri nets are connected into one by a central “Process Manager” and is be used for programming and execution of the code. The programming of the system must be done manually. The product description is not taken into account. [7] present a control architecture for robot cells, which “automatically offers services to the user corresponding to the functionality of the robot cell” in its current setup. Each device in the robot cell brings its own digital description. An automated identification of possible device configurations is realized by describing which interfaces can be attached to each other. The offered skills are used for manual programming. [22] provide a detailed taxonomic framework, which allows modeling of skill primitives of devices with their respective constraints for manufacturing tasks. The work concentrates on task modeling on various levels of detail, from complex assembly tasks to skill primitives. The use case of the presented method is the reusability of knowledge between various robot systems rather than the initial programming and configuration in order to assemble a new product.

The network of devices can be extended by a virtual model of the product. [23] propose a Plug&Produce solution for determining the current state of the devices not only limited to a robot cell, but try to include the rest of the manufacturing system. The skills of the devices and their positioning are taken into account. A virtual representation of the product is used to automatically analyze the feasibility of the manufacturing process. This information is not used to reconfigure or program the devices.

The SIARAS project introduces a methodology for a skill-based comparison between requirements of a product and the skills of a manufacturing system, allowing a product centered task oriented reconfiguration and programming. The description of the manufacturing system and the requirements of the product are entered manually by a human operator and are modeled as an ontology. The appropriate device is discovered by filtering the devices implementing the requested ability and quantitatively comparing the restrictions of requested and offered abilities. Physical reconfiguration of the system itself cannot take place autonomously. A system update during runtime is not possible. [24, 25, 26]
[27] and [28] present a method, which allows the connection of different devices possessing individual skills, represented as holons, to a higher order production system with combined skills. The skills of the production system can be matched with the skill requirements of the product. Based on a capability taxonomy, an adequate device for manufacturing can be selected solution independently. The method is not used for automated programming and configuration of robot cells.

[29] uses the approach of linking product requirements with machine skills to prove the validity of a method for a self-optimizing production control. The requirements and skills are described as “elementary functions “similar to those in [30]. The abilities of the devices update the process duration in real-time and continuously increase the planning accuracy. [31] presents a system, which autonomously matches the various possible sequences of production processes to the available machines. A knowledge based system for altering the production process during run-time in case of disturbances is introduced. Neither [29] nor [31] allow automated physical reconfiguration of devices available in the manufacturing system.

4. Criticism

The related work shows possible partial solutions for the new approach presented in this paper and a variety of methods. However none of the systems integrates an automated extraction of necessary skills from a virtual product description in order to configure and program a robot system. Abilities of the robot system are not used to analyze the feasibility of the production process based on the virtual model of the product. None of the production planning and control systems takes the automated configuration and programming of robot cells into account.

5. Goals

The scope of this paper is the introduction of a methodology, which allows automated programming, configuring and optimization of robot cells by enabling communication between virtual representations of devices in a modular robot cell and the product to be manufactured. Virtual descriptions of the product and the devices need to be developed. Communication, mutual interferences and resulting constraints must be defined. A method for automatically and efficiently generating the virtual model of the product must be introduced. Furthermore a method allowing comparing and optimally assigning product requirements to device abilities must be described. The use case, chosen to explain the methodology, is the automated assembly of products composed of rigid parts in small lot sizes by modular robot cells.

6. Introducing the method

6.1 System overview

In a first step, the part and process data, necessary for virtually describing the entire assembly, is automatically extracted from a CAD file of the fully assembled product (see Fig. 1). The virtual representation of the product to be manufactured is called Augmented-Assembly-Priority-Plan (AAPP) and describes the required production process solution independently. Together with the physical parts of the product, it forms the Cyber-Physical-Product (CPP) (see Fig. 2), a CPS according to [9].

Devices in a robot cell (e.g. an industrial robot, tools, sensors, actuators) have an individual virtual representation of their abilities, constraints and properties and are called Cyber-Physical-Devices (CPD). CPD’s are CPS’s [9]. A Cyber-Physical-Robot-Cell (CPRC) consists of a Centralized-Control-Unit (CCU) and any number of CPD’s (see Fig. 3). The CCU is an IT-System which creates the virtual representation of the robot cell by detecting and analyzing all the available CPD’s in it. It determines possible configurations of the robot cell, corresponding abilities and constraints in each configuration.

In a next step the AAPP and the abilities of the CPRC with the regarding constraints are compared by the CCU, allowing a conclusion about the feasibility of the production process.

If the CPRC meets the AAPP’s requirements, the parts and processes are assigned to the corresponding CPD’s. If more than one configuration of the CPRC comes into question, an optimization algorithm selects the configuration.
which best matches the optimization criteria defined by the user. Finally the CPRC is automatically programmed based on the information from the AAPP.

![Diagram](image1)

Fig. 1. Overview over all subsystems.

![Diagram](image2)

Fig. 2. A Cyber-Physical-Product.

### 6.2. Functional Primitives and Tasks

The solution-independent communication between the various systems is based on the usage of Functional Primitives (FP’s). Exemplary FP’s in assembly are “hold”, “release”, “move”, “measure force” and “detect position”.

FP’s describe the skills of CPD’s. FP’s have quantitative constraints which are dependant of the skills of the individual device. Exemplary the FP “move” from a robot has a three dimensional workspace boundary depending on its kinematics and the FP “hold” from a gripper has a holding force limitation resulting from its design. An important aspect for determining qualitative constraints of available FP’s is the analysis of the mutual interaction of the corresponding CPD’s based on their properties. For example tools connected to a robot and other devices in the CPRC influence the accessibility and collision freedom in the robot cell. Consequently CPD’s need to describe their individual geometry and position. Methods for describing the workspaces of robot systems can be found in [32, 33, 34, 35], but do not take the modularity of a CPRC into account.

FP’s are linked to the control commands of the corresponding CPD. The exemplary FP “move” requests positions and orientations in the coordinate system of the robot in order to execute the movement.

FP’s of two separate devices, which work together, can be logically combined. A robot cell, in which a robot with the FP “move” is connected to a force sensor with the ability “measure force”, can execute the Advanced Functionality (AP) “force controlled movement”. The constraints of the separate FP’s are combined (e.g. the maximum payload of the robot is reduced). The control commands of the two devices are linked to execute the AP. Outside of the CPRC AP’s and FP’s are treated the same (the FP “force controlled movement” could also be offered by a robot with an integrated force sensor). By analyzing all CPD’s available in a CPRC, the CCU determines all available FP’s (see chapter 6.3.2).

FP’s are also used to describe the requirements of the product. Production processes in an AAPP are described as sequences of FP’s. Every qualitative function a FP refers to is connected to corresponding quantitative requirements, e.g. the FP “move” contains a desired start position, an end position and required degrees of freedom; the FP “hold” contains constraints towards the maximum holding force. In order to facilitate the use of knowledge and reduce complexity, recurring sequences of FP’s can be connected to Tasks. An example of a common Task in assembly is
“assemble parts”. It consists of FP’s describing the common process of preparing the handling tool for gripping (“release”), positioning the handling tool (“move”), gripping the part (“hold”), positioning the part against another part (“move”), releasing the part (“release”) and removing the handling tool from the workspace (“move”). Furthermore FP’s have starting and exit conditions, e.g. the “release” FP can only be activated, if the previously requested position was successfully reached (see Fig 4).

The comparison between FP’s offered by a CPRC and FP’s requested by CPP’s is described in chapter 6.3.3. The programming of the CPRC based on the information stored in the AAPP is described in chapter 6.3.5.

6.3. Required Subsystems

In order to reach the before mentioned goals, a new method has to be developed. This chapter explains the necessary operations in and the data exchange between its subsystems.

6.3.1. Automated generation of the AAPP from a CAD file

The initial input for this subsystem is a file, representing the fully assembled product. It contains the geometrical information of every single part and the geometrical constrains between them. Based on the type of constrains and the geometry of the part, the necessary type of Task, build out of a sequence of FP’s, can be determined knowledge-based.

The output of the system is the AAPP, consisting of parts and Tasks (see Fig. 4). The description of the parts in the AAPP includes their geometrical properties, their mass, their initial location and final position in regard to other parts. This part description is used to parameterize the respective FP’s of the previously determined Task. At this stage all information is stored in the coordinate systems of the corresponding parts. Only primary production processes [36] can be extracted from the CAD-File. The AAPP describes all feasible assembly orders.

6.3.2. Analysis of the skills of a Cyber-physical robot cell (CPRC)

In order to allow an automated comparison between the requirements of the products and the full skills offered by the manufacturing system, a description of the latter needs to be generated. The here under mentioned approach allows a detailed and flexible real-time analysis of the current state of the manufacturing system using the potential of Cyber-Physical-Systems.

After detecting all CPD’s in the robot cell and downloading their virtual representations, the CPRC begins the determination of all possible configurations. Every CPD has a digital description of its physical interfaces (e.g. digital, fluidic, mechanical, and electrical). These are divided in input and output interfaces. Input interfaces describe the prerequisites that have to be met in order for the CPD to be fully functional. Output interfaces of a CPD
are used to provide the necessary prerequisites to further CPD’s. The output interfaces a CPD can provide can be dependent on the successful connection of the required interfaces (e.g. a robot can only supply a gripper with compressed air, if it is connected to a compressed air supply). The CPD detects autonomously which FP’s and interfaces it can provide by analyzing which input interfaces were successfully connected. The virtual representation of the CPD contains the logical conditions between the input and output interfaces (see Fig. 5). By analyzing all interfaces of the CPD’s in the robot cell, the CPRC can completely determine its possible configurations.

The abilities of CPD’s are represented in form of FP’s. The usability of its FP’s is, like the output interfaces, dependent on the successful connection of input interfaces. An electrical gripper can only “hold” a part, if it is connected to an electrical energy source. Based on the previously determined possible configurations, the CPRC can now qualitatively and quantitatively analyze which full sets of FP’s with corresponding constraints are available in each configuration (see chapter 6.2).

The output of the subsystem is the virtual representation of all possible functional primitives with the corresponding constrains a CPRC can execute.

6.3.3. Skill comparison

A core element of the product-centered programming and configuring of the manufacturing system is the comparison and matching between the required and offered skills.

The set of configurations with its corresponding FP’s provided by the CPRC has to be matched with the set of possible sequences of FP’s represented in the AAPP in order to assign parts and assembly processes unambiguously to CPD’s available in the robot cell.

In a first step a qualitative comparison takes place. The CPRC analyzes autonomously, if it can offer a single configuration in which all FP’s required by the product are available. If this is not the case, all configurations are analyzed in order to determine whether the qualitative requirements of the product can be met with a reconfiguration of the CPRC during the assembly process.

After choosing one or more applicable configurations, the quantitative comparison of the required and offered FP’s takes place. Functional primitives are connected to Comparison Modules (CM). These are responsible for determining whether the product requirements fit in the process window of the CPRC or exceeds it. Exemplary CM’s are “accessibility testing”, “collision testing” and “safe gripping testing”. [33, 34, 35, 36] present methods for accessibility and collision testing in robot cells, but need to developed further in order to take the modularity of a CPRC into account.

Until now, the positions of the parts in the AAPP are solely described in the coordinate systems of the parts. For “accessibility testing” and “collision testing” the initial and final position of the parts to be picked, moved and assembled has to be known in the coordinate system of the robot. The initial positions of all parts can be determined with adequate sensors. Alternatively the position of feeding devices and fixtures of the parts in the robot cells may be known to the system, reducing the sensory effort. After analyzing which CPD’s are able to hold the parts, possible positions and orientations of the parts in the robot cell are known. If all initial and final positions of the parts are determined, the before mentioned accessibility and collision testing comparison modules can be run successfully. The positions of the parts in the coordinate system of the robot are stored in the AAPP.
If information which is necessary for the comparison is not available, e.g. “safe gripping testing” is supposed to be executed and the mass of the part is unknown, the system autonomously queries the user, e.g. the product designer, and calls for the required input. The output of this subsystem is a set of suitable linkages between the FP’s offered by the CPRC and the FP’s required by the CPP.

Due to the fact that the AAPP is created from a CAD-File, the comparison between offered skills and those required by a product yet only existing in a digital description is possible. Product requirements which can’t be met by the current production system are determined. Either corresponding product details are changed or an investment in CPD’s satisfying the product requirements is induced. The presented functionality would reduce the number of iterations between product designers and process planners.

6.3.4. Selection of the optimal configuration

After determining those CPRC configurations, which passed the comparison with the required skills of the product, the remaining configurations are evaluated based on optimization criteria defined by the user.

In a scenario with more than one CPRC and a variety of products to assemble, production planning and control is taken to a new level of flexibility and transparency. The planning and control algorithm can take all available configurations of the production system and all possible production sequences of the products into account, greatly increasing its flexibility. The measurement of utilization is no longer bound to physical resources, but can be determined solely on the fine grained level of solution independent FP’s.

The output of this subsystem is an unambiguous linkage between the FP’s offered by the CPRC and the FP’s required by the CPP. Each part and process of the product to be assembled is assigned to components of the robot cell.

6.3.5. Programming the CPRC

The programming of the CPRC takes place, if all necessary variables in the AAPP are known and the optimal configuration was chosen. The information from the functional primitives from the AAPP is send to the functional primitives of the CPD’s. A solution-independent, task oriented programming takes place.

7. Conclusion

The presented system shows the possibilities of modular robot systems equipped with smart devices and smart products. The continuous use of solution independent skill descriptions in form of functional primitives for determining product requirements, determining robot system skills, analyzing the feasibility of the assembly,
optimizing the assembly and programming the robot cell allows a variety of new benefits. Due to the precise description of the constraints of the skills of the robot cell the feasibility of the assembly process can be determined with certainty. If necessary, the product or the robot cell can be changed. The robot system is programmed and configured autonomously. A new level of detail, actuality and flexibility for production planning and control can be achieved. We are currently further specifying the functional primitives, which form the fundament for the communication between all subsystems. A feasible extraction of FP’s from CAD files and the logic of mutual interactions between FP’s are being researched. We are currently building a prototypical demonstrator, allowing the execution of the presented methodology.

Acknowledgements

The presented work was conducted as part of the research project CyProS [37].

References

[1] Reinhart G, Krug S. Robotersysteme in der Kleinserie – Effizienz von der Planung bis zum Einsatz. iwb Seminarberichte 91, München, Utz 2009, p. 1-16.
[2] Abele E, Reinhart G. Zukunft der Produktion. München: Carl-Hanser-Verlag; 2011.
[3] Wiendahl H-P, ElMaraghy HA, Nyhuis P, Zäh MF, Wiendahl HH, Duffie N, Brieke M. Changeable Manufacturing – Classification, Design and Operation. CIRP Annals-Manufacturing Technology, Annals of the CIRP 2007;6/2.
[4] Anderl R, Eigner M, Sendler U, Stark R. Smart Engineering – Interdisziplinäre Produktenstehung. In: Reiner Anderl, editor. acatech Diskussion April 2012, acatech – Deutsche Akademie der Technikwissenschaften. Technische Universität Darmstadt; 2012.
[5] Krug S. Automatische Konfiguration von Robotersystemen. München: Herbert Utz Verlag GmbH; 2013.
[6] Reinhart G, Hüttner S, Krug S. Automatic Configuration of Robot Systems – Upward and Downward Integration. In: Reinhart G. et al. (Hrsg.): Proceedings of the 4th International Conference on Intelligent Robotics and Applications (ICIRA 2011), Aachen, 2011. Heidelberg: Springer;2011. p. 102-111.
[7] Naumann M, Wegener K, Schraft RD. Control Architecture for Robot Cells to Enable Plug’n’Produce. 2007 IEEE International Conference on Robotics and Automation 2007.
[8] Geisberger E, Broy M. Integrierte Forschungsagenda Cyber-Physical Systems. Agenda CPS acatech Studie March 2012. München: Dr. Eva Geisberger fortiss GmbH; 2012.
[9] Reinhart G, Engelhardt P, Geiger F, Philipp TR, Wahlster W, Zühlke D, et al. Cyber-Physische Produktionssysteme. wt-online 2-2013 Sonderheft Industrie 4.0 2013:84-89.
[10] Gu P, Yan X. CAD-directed automatic assembly sequence planning. International Journal of Production Research 1995;33/11:3069-3100.
[11] Bley H, Bossmann M. Standardisierte Produktdaten für die automatisierte Montageplanung Featurebasierte Montageplanung unterstützt den Simultaneous-Engineering-Prozess. wt Werkstatistik online 2005;9.
[12] Rührdanz F, Mosemann H, Wahl F. Generating and evaluating stable assembly sequences. Advanced Robotics 1997;11/2:97-126.
[13] Swaminathan A, Barber KS. An experience-based assembly sequence planner for mechanical assemblies. IEEE Trans Robot Autom 1996;12/2:252-266.
[14] Yin ZP, Ding H, Li HX, Xiong YL. A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Computer Aided Design 2003;35/1:37-56.
[15] Fan J, Dong J-X. KVAS- a knowledge-based virtual assembly system. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics 2001:1041-1046.
[16] Li-Ming Ou, Xun Xu. Relationship matrix based automatic assembly sequence generation from a CAD model. Computer-Aided Design 2013;45:1053-1067.
[17] Mosemann H, Wahl FM. Automatic Decomposition of Planned Assembly Sequences Into Skill Primitives. IEEE Transactions on robots and Automation 2001;17/5.
[18] Thomas U, Wahl FM. A System for Automatic Planning, Evaluation and Execution of Assembly Sequences for Industrial, Robots Intelligent Robots and Systems. Proceedings IEEE/RSJ International Conference on 2001;3:1458-1464.
[19] Mosemann H. Beiträge zur Planung, Dekomposition und Ausführung von automatisch generierten Roboteraufgaben. PhD thesis. Aachen: Shaker Verlag; 2000.
[20] Arai T, Aiyama Y, Maeda Y, Sugi M, Ota J. Agile Assembly System by “Plug and Produce”. Annals of the CIRP - Manufacturing Technology 2000;49/1.
[21] Mendes JM, Leitão P, Colombo AW, Restivo F. High-level Petri nets for the process description and control in service-oriented manufacturing systems. International Journal of Production Research 2012;50/6:1650-1665.
[22] Huckaby J, Christensen HI. A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics. AAAI Workshop - Technical Report Volume WS WS-12-06 2012:94-101.
[23] Keddis N, Kainz G, Buckl C, Knoll A. Towards Adaptable Manufacturing Systems. Proceedings of the 2013 IEEE International Conference on Industrial Technology 2013.

[24] Bengel M. Workpiece-centered Approach to Reconfiguration in Manufacturing Engineering. Heimsheim: Jost-Jetter Verlag; 2010.

[25] Malec J, Nilsson A, Nilsson K, Nowaczyk S. Knowledge-Based Reconfiguration of Automation Systems. Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering 2007.

[26] Malec J, Nilsson K, Bruyninckx H. Describing Assembly Tasks in Declarative Way. Proceedings Workshop on Semantics, Identification and Control of Robot-Human-Environment Interaction 2013.

[27] Nylund H, Salminen K, Andersson PH. Digital virtual holons - An approach to digital manufacturing systems. 2008 Manufacturing Systems and Technologies for the New Frontie. The 41st CIRP Conference on Manufacturing Systems 2008:103-106.

[28] Salminen K, Nylund H, Andersson P. Role based Self-adaptation of a robot DiMS based on system intelligence approach. Flexible Automation and Intelligent Manufacturing 2009.

[29] Lau J. Methodik für eine selbstoptimierende Produktionssteuerung. München: Herbert Utz Verlag; 2010.

[30] VDI Richtlinie 2860: Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole. Düsseldorf: VDI-Verlag; 1990.

[31] Ostgathe M. System zur produktbasierten Steuerung von Abläufen in der auftragsbezogenen Fertigung und Montage. München: Herbert Utz Verlag; 2012.

[32] Goyal K, Sethi D. An Analytical Method to Find Workspace of a Robotic Manipulator. In: Journal of Mech. Engineering 2010;41/1:25-30.

[33] Alicatore D. Determining Manipulator Workspace Boundaries using the Monte Carlo Method and Least Squares Segmentation. Proceedings of 23rd ASME Mechanisms Conference 1994.

[34] Miller K, Petitt JD. Six-Dimensional Visualisation of End-Effector Pose Using Colour Spaces. Proceedings Australasian Conference on Robotics and Automation 2002:216-221.

[35] Zacharias F. Knowledge Representations for Planning Manipulation Tasks. Diss., Technische Universität München 2011.

[36] Lotter B. Montage in der industriellen Produktion: ein Handbuch für die Praxis. Berlin: Springer; 2006.

[37] www.projekt-cypros.de