Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 44 (2016) 44 – 49

www.elsevier.com/locate/procedia

6th CIRP Conference on Assembly Technologies and Systems (CATS)

# CAD-based automated assembly planning for variable products in modular production systems

Joachim Michniewicz[a]*, Prof. Dr.-Ing. Gunther Reinhart[a], Dr. Stefan Boschert[b]

*[a]Institut für Werkzeugmaschinen und Betriebswissenschaften, Technische Universität München, Boltzmannstr.15, 85748 Garching, Germany*
*[b]Siemens Corporate Technology, Research and Technology Center, Otto-Hahn-Ring 6, 81739 Munich, Germany*
**Corresponding author. Tel.: +49 (0) 89 289 15476; fax: +49 (0) 89 289 15555 E-mail: Joachim.Michniewicz@iwb.tum.de*

**Abstract**

The paper presents an approach allowing to utilize the flexibility of modular and reconfigurable production systems for the automated assembly of multi-variant products. Required assembly processes as well as valid assembly orders are automatically extracted from the CAD-file of the individual product. Assembly processes are described as skills and are automatically assigned to capable production resources after a simulative verification, generating an optimal assembly plan. The described planning process takes the properties and geometry of production resources, their mutual interactions as well as the layout and feasible material paths of the entire production system into account.

*Keywords:* assembly, computer aided design (CAD), mass customization, production planning

## 1. Introduction

Currently the manufacturing industry is facing an increasing demand for flexibility due to the trend of mass customization. The amount of product variants ascends, life cycles fluctuate increasingly and lead times need to be minimized [1],[2].
Despite that, current factories are planned and optimized for specific products or specified product families. The introduction of a new product variant or the integration of a new resource to the production system requires high manual effort and is time consuming [3].
Mass customization and creation of variants is usually achieved during product assembly [4]. Increasing modularization and compatibility of product components enlarges the product range and facilitates the introduction of new product variants and new components. Assembly planning and the selection of the optimal resource configuration however becomes increasingly complex and time consuming [5]. High manual effort is required to validate the practicability of the required production processes with the available resources as well as the selection of an optimal allocation of the resources to the necessary assembly steps [6].
Automated assembly systems are usually used for specific pre-defined tasks. Only a narrow spectrum of the functional range of resources is utilized (e.g. the workspace of a robot in comparison to a repetitive motion or the restriction of a gripper to one gripping point of one specific part) due to manual efforts for planning and validation of the required processes. Furthermore the control code for the entire production system needs to be developed, tested and implemented manually [7].
In order to facilitate the determination of product requirements towards necessary assembly processes, different approaches for the automated analysis of CAD-Models of the product assembly have been introduced [8-12]. However, the presented methods do not take the generation and selection of an optimal assembly plan for a modular and reconfigurable production systems into account. The solution space, constraining alternative possible allocations of the required processes to available resources, varying material flows as well as different valid assembly orders are not taken into consideration sufficiently.
Methods for automated production planning in flexible production systems depend upon various manual inputs in

doi:10.1016/j.procir.2016.02.016

order to define the process sequence required by the product in the specific description language of the system [13-18]. The effort necessary for creating each product-individual model in production with small batch sizes is not taken into account. Usually only one valid process sequence is given, disregarding the flexibility of the product to different feasible assembly sequences. The factory topology has to be entered manually in a specific modelling language, further increasing required expert knowledge and planning efforts in flexible production systems.

The presented paper suggests a skill-based approach for solving these problems. The overall system has been introduced in detail in previous publications [19-21]. The system minimizes adaptation effort due to common software tools widely used for input of product (CAD software) and production system data (programmable 3D factory simulation), which are then used to automatically generate the specific models needed for further computation. The focus of this paper are the efficient automated generation of the product requirements from CAD-Data taking the factory layout into account as well as a detailed validation of the overall system.

The paper is organized as follows: section 2 gives a brief overview of the system architecture and functionality. Section 3 specifies the methodology behind the automated CAD-File analysis. Section 4 contains the validation of the overall system. Section 5 discusses achieved increases in flexibility and adaptability. Finally, section 6 summarizes the paper and provides an outlook on future work.

## 2. System architecture and functionality

### 2.1. Tasks and Functional Primitives

The basis for the flexible allocation of resources to required assembly processes are *Tasks* and *Functional Primitives (FP)*, allowing a solution independent representation of requirements and abilities. They are stored in an extensible library. Elemental processes like movement, gripping and releasing are modeled as FP, which are derived from a widely accepted guideline [22]. Tasks represent more complex recurring processes often found in assembly like “assembly of two parts”, “screwing” or “handling of a part”. Tasks are modeled as sequences of FP. The sequence of FP in a Tasks represents logically linked actions like the alternating closing and opening of grippers as well as movement while handing a part from one robot to another (see Fig. 1). However the FPs are parametrized individually for each product-specific sub-step during assembly.

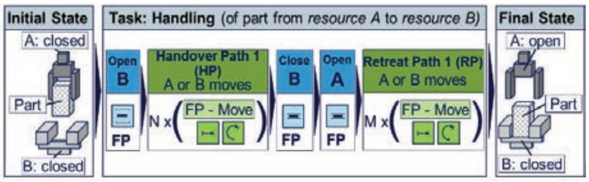


Fig. 1: Functional Primitives contained in the Task "Handling”. *Handover Path* describes the movement during the approach of the resources, while *Retreat Path* describes the collision free retreat movement after the handling operation.

### 2.2. Modeling of the production system

In order to facilitate the search and allocation of feasible resources, a model based on functional primitives representing the overall abilities as well as feasible material paths through the production system was introduced [19-21]. The model is called *Production Graph (PG)* (see Fig. 2d) and is automatically generated from digital models of available resource, placed in a 2D- or 3D simulation model of a production system (see Fig. 2b). In order to facilitate the use of the system, the PG is generated from unaltered simulation files of common simulation programs. Consequently changes to the production system are performed easily by adding, removing or moving resources in a familiar interface. The PG is updated automatically.

The digital model of each resource contains individual representations of the properties, abilities (stored as FPs), constrains and its geometrical workspace. Digital models for different grippers, robots, conveyors and screwdrivers are implemented.

Vertices in the PG represent workstations with their corresponding abilities represented as Functional Primitives, which result from the individual resources available in a workstation. Edges represent valid material flows between vertices. Vertices and edges are generated automatically by detecting overlaps between workspaces of the different resources (e.g. overlapping workspaces of two robots generate an edge; a robot workspace overlapping reachable tool generates a vertex).

### 2.3. Modeling of the product requirements

The product requirements as well as feasible assembly orders are modeled as a directed graph called the *Augmented Assembly Priority Plan (AAPP)*. The AAPP consists of vertices, representing Tasks, and edges, modeling feasible assembly orders. Tasks contain a sequence of FPs. Value adding Tasks like “assembly” or “screwing” consist of two initial subassemblies which are joined together to form one new subassembly. The output of the final Task is the finished product (see Fig. 2c).

The generation of the product requirements from CAD files (see Fig. 2a) is explained in detail in section III. Without knowledge about the production system, only value adding assembly processes can be extracted from CAD-Files of the product to be assembled. In this state AAPP is solution neutral. The combination of the PG and the solution neutral AAPP in order to allocate all resources optimally and generate the factory specific AAPP is briefly introduced in the next part.

### 2.4. Automated allocation of resources

In order to generate a valid assembly plan, containing the optimal allocation of resources to assembly processes required by the product, initially all value adding processes are considered. For the two initial subassemblies in each Task pairs of adjacent vertices in the PG are searched, assigned and added to the Task description in the *factory specific AAPP*, following the assumption that both subassemblies must be held by individual resources whose workspaces overlap. This allows the feasibility test of all value adding processes as well as the detection of missing or insufficient abilities.

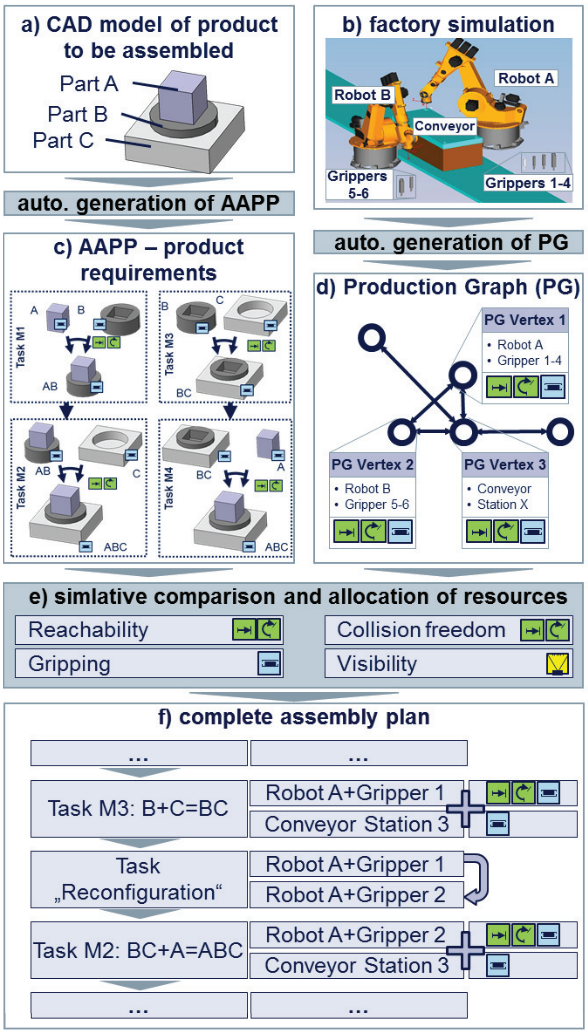


Fig. 2: system overview.

In a next step feasible material flows along the edges of PG are generated by linking the valid consecutive value adding Tasks. Secondary Tasks, representing necessary handling operations, are allocated to specific resources and are added to the factory specific AAPP. Secondary Tasks like “handling” are not value adding and do not change the product geometry. They therefore only consist of one unaltered subassembly.

In order to securely allocate the value adding and secondary Tasks, multiple simulative feasibility test are performed (see Fig 2e). Exemplary implemented simulative tests address reachability, collision freedom, gripping, screwing and visibility of parts for cameras. When all value adding Tasks are linked to feasible resources and all secondary Tasks (also linked to feasible resources) are generated, one or more valid assembly plans are successfully generated and stored in the factory specific AAPP. Finally, the complete assembly plan with the shortest overall duration is selected (see Fig. 2f).

## 3. Generation of product requirements from CAD-Files

CAD data of product assemblies in all common software tools and CAD data types consists of the geometrical models of all parts in the product and can furthermore contain geometrical constrains between these parts. An unaltered CAD model of the product contains no information regarding feasible assembly orders or required assembly processes, not to mention necessary Tasks or Functional Primitives. Different methods, presented in the following sections, need to be applied in order to generate the required solution independent AAPP efficiently. The method of virtual “assembly by disassembly”, during which all parts are incrementally moved in order to generate a collision free path until a certain distance is reached, is commonly known [8-12]. It becomes highly inefficient without a pre-sequencing and efficient sorting of the parts. These steps are necessary to minimize the amount of failed and prematurely reattempted virtual disassemblies. This reduces the time between the design of a new product and the validation of the feasibility of the assembly sequence, allowing quicker redesigns of the product or the production system.

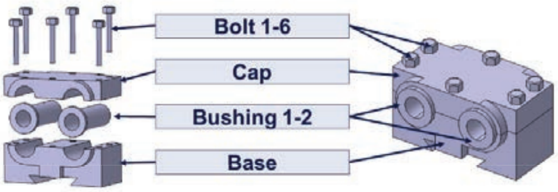


Fig. 3: exemplary product.

During the generation of product requirements from CAD-Data, each assembly sequence being generated is stored in a matrix, containing all parts of the product. Parts in the top left corner of the matrix are first to be disassembled and consequently the last to be assembled. The final matrices are then transformed into the graph based AAPP. The tree structure of the graph contains the valid assembly sequence. The information regarding necessary assembly movements and final positions of the parts are stored in the AAPP by parametrizing its Functional Primitives and Tasks.

The following functions can be implemented in any CAD-tool providing a sufficient application programming interface. The standard functions of present CAD tools utilized in the following steps are movement, hiding and showing of parts, reading constrains and positions of parts as well as collision detection.

### 3.1. Pre-Sequencing – Generation of rough assembly order

Before the “assembly by dissassembly” sequence is initiated, users can manually enter which parts or part families shall be incrementally moved first or last, automatically putting these parts in the first or last row of the matrix. In a next step, the user is requested to input a main mounting direction, valid for the entire product. Many products designed with assembly requirements in mind have a single mounting direction, especially when following a layered design or when all parts are mounted into one base part [6].

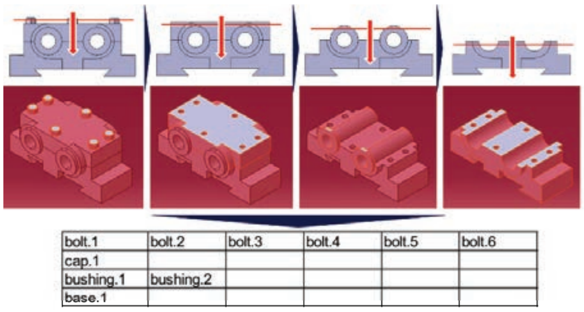
A plane is generated outside of the product model and moved in the mounting direction. Successive collisions between parts and the plane are detected and the parts written into the matrix, generating the rough assembly order. The assumption behind this method is that parts on the outside of the product, colliding with the plane first, also need to be disassembled first. By prioritizing different parts or entering different main mounting directions, multiple assembly orders can be generated (see Fig. 4).

### *3.2. Assembly by disassembly*

The multiple matrices generated in the previous step are validated by incrementally moving the parts of the product in the specified order. A part is moved until it reaches a predefined distance to the remaining subassembly. The movement direction is determined by geometrical constrains between the parts, the main mounting direction as well as the coordinate systems of the individual parts. In order to further increase efficiency, connecting elements like bolts or screws are moved primarily.

If a part can't be removed due to collisions, for each movement direction the other parts it collided with are stored. The parts are only moved again in an obstructed direction, if all stored parts are removed, following the assumption that all obstructive parts in the movement direction need to be disassembled first. The goal is to maximize sorting efficiency for the remaining parts.

The assembly movements of each part is stored and used to parametrize the FPs "move". The analysis of the product is finished, when all parts have been successfully digitally removed.



| | | | | | |
|---|---|---|---|---|---|
| bolt.1 | bolt.2 | bolt.3 | bolt.4 | bolt.5 | bolt.6 |
| cap.1 | | | | | |
| bushing.1 | bushing.2 | | | | |
| base.1 | | | | | |

Fig. 4: Visualization of the plane moving through the exemplary product. Contacted parts are blended out and stored into the matrix successively.

### *3.3. Input of part interfaces*

In order to perform an assembly step, both subassemblies must be held by a resource like a fixture, gripper or screwdriver. In the current development status, the position of interfaces of each part as well as interface requirements must be entered manually by the user, following the template of the FP "hold" or "screw". This parametrizes the solution independent Task required by the product. Certain entries like "mass" or in case of screws "thread type" or "length" can be automatically extracted from the CAD model of the part.

### *3.4. From assembly sequences to product requirements (AAPP generation)*

The complete AAPP is generated by modeling the assembly of each consecutive part as a Task which contains the sequence of earlier parametrized FPs. For every generated assembly order an individual graph is generated, all of which are summarized in one AAPP.

### *3.5. Use of Production Graph for better CAD-analysis results*

In order to further optimize the automated generation of the AAPP, the Production Graph can be utilized. Usually production systems have a defined material flow direction. The vertices of a PG are linked by directional edges, representing valid material flows. By analyzing the successive vertices of the PG, sequences of executable FPs without backflow of material can be generated. This sequence of FPs is then used to prioritize parts for the assembly by disassembly strategy. This is efficient especially when comparable products have already been successfully assigned to the production system.

## 4. Application example of the overall system

In order to evaluate the overall system, this section illustrates its application for the assembly planning of three different exemplary products in regard to a reconfigurable production system.

### *4.1. Exemplary products*

The three exemplary products a, b and c are housings mounting different numbers of bushings, displayed in Figure 5a, 5b and 5c. All parts are rigid and are not deformed during assembly. The automatically generated solution independent AAPPs are shown in Fig. 5d, 5e and 5f. One assembly sequence per product was generated and contains all required assembly steps. Product b has wider base part.

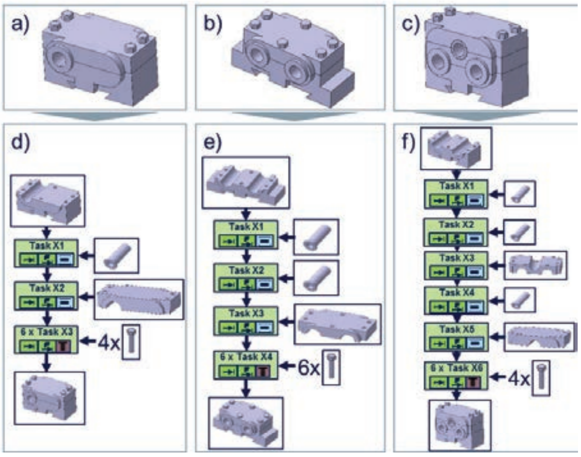


Fig. 5(a,b,c): Exemplary products a, b and c with different part numbers and types. Figures (d), (e) and (f) show the automatically generated AAPP.

### *4.2. Exemplary production system*

The simulated production systems consists of three robots aligned along a conveyor, see Fig. 6. The first robot is responsible has tools allowing it to hold cylindrical parts (C1). The second robot has two configurations. It can be equipped with two different grippers (P1 and P2) suitable for holding parts with parallel surfaces. The third robot is equipped with an electric screwdriver.

The parts on the conveyor belt can be stopped at 3 positions. The PG shown in Fig. 7 is generated automatically from the simulation model.

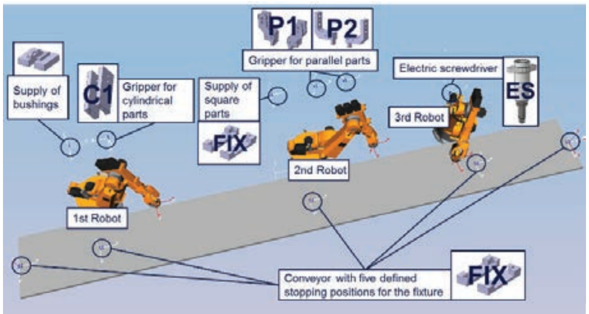

P1
P2
Gripper for parallel parts
Electric screwdriver
ES
Supply of bushings
C1
Gripper for cylindrical parts
Supply of square parts
FIX
3rd Robot
2nd Robot
1st Robot
Conveyor with five defined stopping positions for the fixture
FIX

Fig. 6: Image of the simulated production system with highlighted resources.

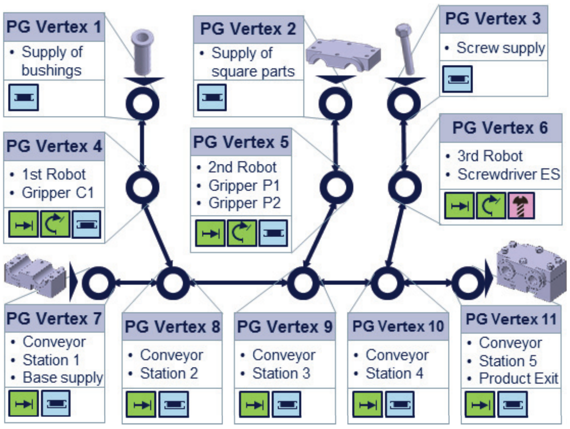


Fig. 7: resulting Production Graph for the simulated production system.

### 4.3. Results

Fig. 9 displays the automatically generated complete assembly plan (factory specific AAPP) for product a, containing the valid allocation of all required value adding Tasks from the solution independent AAPP as well as all necessary secondary Tasks, generated individually for the production system. All Tasks are successfully allocated to resources in the production system.

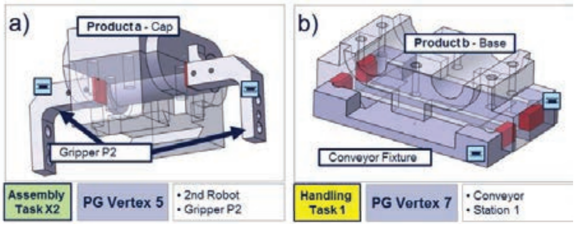


Fig. 8: (a) detected collision (highlighted red) between product a and Gripper P2 during the assembly of the Cap leads to an exclusion of the configuration “2nd Robot with Gripper P2” for the assembly plan; (b) The wider Base-part of product b causes a collision with the fixture of the conveyor, rendering the generation of an assembly plan for part b for the production system impossible.

The 2nd configuration of the PG Vertex 2 (2nd Robot with gripper P2) was excluded from the assembly plan for product a due to collisions detected during simulation (see Fig. 8a).

For product b no valid assembly plan could be generated due to the wider base part. This detected non-fulfilment is displayed, allowing a quick adaptation to the product or the production system (see Fig. 8b).

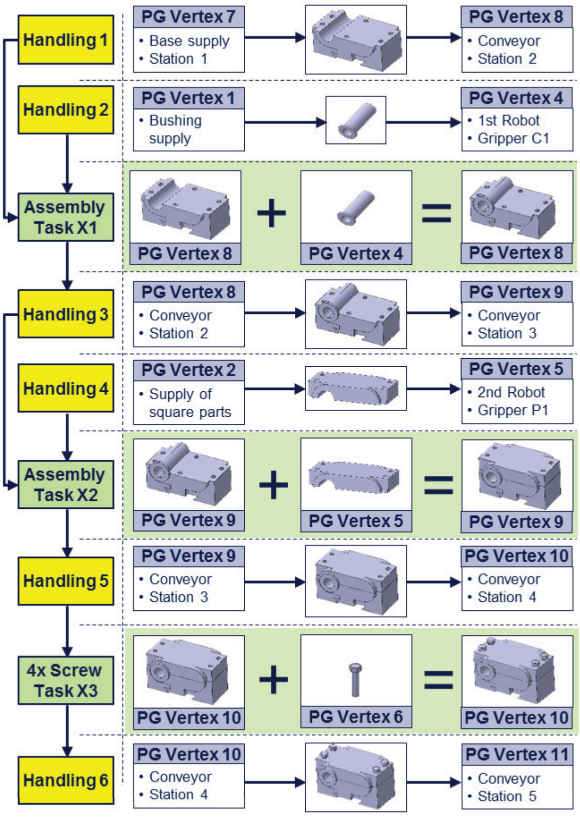


Fig. 9: automatically generated assembly plan (factory specific AAPP) for product a, containing all necessary value adding (green) and secondary Tasks (yellow).

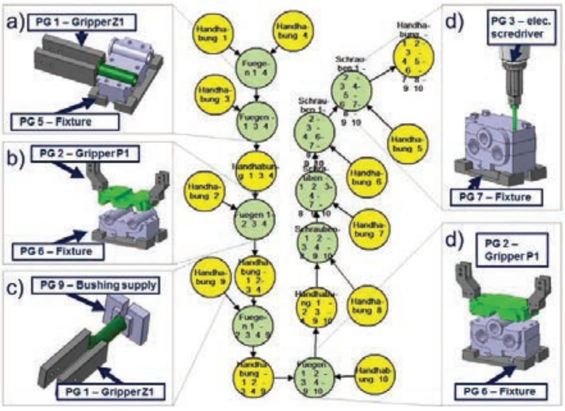


Fig. 10: The graph represents the valid assembly plan for product c. Selective simulation steps are displayed and linked to the respective Tasks in the graph. Value adding Tasks are green, secondary Tasks are yellow.

For product c a valid assembly plan was generated, containing nine value adding and fifteen secondary Tasks (see Fig. 10). The presented results of the automated assembly planning are valid in regard to the executed simulative tests which were performed, namely gripping, collision freedom, robot reachability and screwing. The comparison of requirements and capabilities for each FP analyzes predefined properties. Consequently exclusion criteria not contained in the model are not considered. Products and resources are modeled as rigid parts. Deformations are neither taken into account during the generation of the product requirements nor during the simulation of the assembly processes.

## 5. Summary and outlook

In this paper we present an automated assembly planning system for modular production systems based on solution independent Functional Primitives. The system allows automated validation and planning of assembly processes with unprecedented flexibility and ease. The automated generation and update of the model of product requirements as well as available abilities in a production system from commonly used digital models with familiar user interfaces promises easy integration and high usability. Necessary secondary processes are detected automatically. The simulative execution of all required value adding and secondary processes takes different assembly sequences of the product to be assembled as well as different configurations and material flows of the production system into account. The feasibility of an assembly with the available resources is validated, the optimal assembly sequence chosen and allocated to the available resources automatically. The automated determination and display of non-fulfilment of product requirements accelerates the adaptation of the product or the production system respectively.

The system takes different flexibilities of the production system into account, increasing the useful life and utilization of the available resources. Furthermore iterative manual non-productive adaptations of products or the production system are eliminated, reducing time to market and accelerating introduction of new products. An unprecedented level of automation during assembly planning is reached, based on commonly available data and software tools.

The next step in our work is the implementation of further Functional Primitives in order to cope with a bigger variety of products (e.g. “measure force” or “welding”). Furthermore elastic elements of parts like clips, which can often be found in assembly, need to be taken into account during the assembly-by-disassembly method. In order to further reduce the required manual inputs, an automated generation of valid gripping points is planned. The approach has to be evaluated on larger problems to determine its efficiency for large industrial setups.

## References

[1] S. Hu, X. Zhu, H. Wang, and Y. Koren, “Product Variety and Manufacturing Complexity in Assembly Systems and Supply Chains,” CIRP Annals-Manufacturing Technology, vol. 57, no. 1, pp. 45–48, 2008

[2] Abele, E.; Reinhart, G.: Zukunft der Produktion. Herausforderungen, Forschungsfelder, Chancen. München: Hanser, Carl 2011. ISBN: 3446425950.

[3] M.-F. Zah, M. Beetz, K. Shea, G. Reinhart, O. Stursberg, M. Ostgathe,¨ C. Lau, C. Ertelt, D. Pangercic, T. Ruhr¨ et al., “An Integrated Approach to Realize the Cognitive Machine Shop,” in Proceedings of the 1st International Workshop on Cognition for Technical Systems, 2008

[4] Reinhart, G.; Zäh, M. F.: Intelligente und effiziente Automatisierung der Montage. wt Werkstattstechnik 97 (2007)

[5] Westkämper, E.: Research for Adaptive Assembly: First CIRP International Seminar on Assembly Systems. 15.-17. November. Stuttgart: Fraunhofer-IRB-Verl. 2006, ISBN: 978-3-8167-7213-2.

[6] LOTTER & WIENDAHL 2012 Lotter, B.; H.-P. Wiendahl: Montage in der industriellen Produktion Ein Handbuch für die Praxis. ger. 2. Aufl. 2013. VDI-Buch. Berlin, Heidelberg

[7] Reinhart G, Krug S. Robotersysteme in der Kleinserie - Effizienz von der Planung bis zum Einsatz. iwb Seminarberichte 91, München, Utz 2009

[8] Li-Ming Ou, Xun Xu. Relationship matrix based automatic assembly sequence generation from a CAD model. Computer-Aided Design 45 (2013)

[9] Mosemann H, Wahl FM. Automatic Decomposition of Planned Assem bly Sequences Into Skill Primitives. IEEE Transactions on Rob ots and Automation, Vol. 17, No. 5, Octob er 2001

[10] Thomas U, Wahl FM. A System for Automatic Planning, Evaluation and Execution of Assembly Sequences for Industrial, Rob ots Intelligent Robots and Systems. Proceedings. 2001 IEEE/RSJ International Conference on (Volume:3 ), pp. 1458- 1464

[11] Mosemann H. Beitrge zur Planung, Dekomposition und Ausfhrung von automatisch generierten Roboteraufgab en. ISBN 3-8265-7710-8, Shaker Verlag, Aachen 2000, Dissertation

[12] Alfadhlani; Samadhi, T. M. A. Ari; A. Ma’ruf; I. S. Toha: Automatic Collision Detection For Assembly Sequence Planning Using A Three-Dimensional Solid Model. en. Journal of Advanced Manufacturing Systems 10 (2011) (02), S. 277–291. ISSN: 0219-6867. DOI: 10. 1142/S021968671100220X

[13] Keddis, N., Kainz, G., & Zoitl, A.: Capability-based planning and scheduling for adaptable manufacturing systems. In Emerging Technology and Factory Automation (ETFA), 2014 IEEE (pp. 1-8)..

[14] Bengel M. Workpiece-centered Approach to Reconfiguration in Manufacturing Engineering. 2010, Jost-Jetter Verlag He im sheim, ISBN 3-939890-60-X

[15] W. Lepuschitz, B. Groessing, M. Merdan, and G. Schitter, “Evaluation of a Multi-Agent Approach for a Real Transportation System,” in IEEE International Conference on Industrial Technology (ICIT), 2013. IEEE, 2013, pp. 1273–1278.

[16] Y. Alsafi and V. Vyatkin, “Ontology-based Reconfiguration Agent for Intelligent Mechatronic Systems in Flexible Manufacturing,” Robotics and Computer-Integrated Manufacturing, vol. 26, no. 4, pp. 381–391, 2010.

[17] Backhaus, J., and G. Reinhart. "Digital description of products, processes and resources for Task-oriented programming of assembly systems." Journal of Intelligent Manufacturing (2015): 1-14.

[18] Stenmark, Maj, and Jacek Malec. "Knowledge-based instruction of manipulation Tasks for industrial robotics." Robotics and Computer-Integrated Manufacturing 33 (2015): 56-67.

[19] Michniewicz, Joachim, and Gunther Reinhart. "Cyber-physical Robotics– Automated Analysis, Programming and Configuration of Robot Cells based on Cyber-physical-systems." Procedia Technology 15 (2014): 567-576.

[20] Michniewicz, Joachim, and Gunther Reinhart. "Cyber-Physical-Robotics– Modelling of modular robot cells for automated planning and execution of assembly Tasks." Mechatronics (2015). doi:10.1016/j.mechatronics.2015.04.012

[21] Michniewicz, Joachim and Gunther Reinhart. “Cyber-Phyische System in der Robotik – Automatische Planung und Durchführung von Montageprozessen durch Kommunikation zwischen Produkt und Produktionssystem“ in „Intelligente Vernetzung in der Fabrik“, Fraunhofer Verlag (2015), ISBN 978-3-8396-0930-9, pp. 229-262

[22] Richtlinie, V. D. I. "2860: Montage-und Handhabungstechnik." VDI, Düsseldorf (1990