Mechatronics xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks

Joachim Michniewicz *, Gunther Reinhart

Institute for Machine Tools and Industrial Management, Boltzmannstr. 15, D-85748 Garching, Germany

ARTICLE INFO

Article history:
Received 15 October 2014
Revised 9 March 2015
Accepted 19 April 2015
Available online xxxx

Keywords:
Cyber-Physical-Systems
Reconfiguration and programming of robot systems
Task description language
Assembly
Computer aided design
Production planning

ABSTRACT

This paper presents an approach, which allows the utilization of the entire flexibility of modular robot cells. Solution independent requirements of products to be assembled are automatically extracted from their individual CAD files and used for the optimal selection and task oriented programming of cooperating devices in the robot cell. These devices and the products to be manufactured are Cyber-Physical-Systems (CPS), which possess individual virtual models of their requirements and abilities. Based on the communication between the CPS and the fusion of their models, the feasibility of the required assembly tasks with the devices available in the robot cell is determined. The product description is used to optimally assign the required assembly processes to cooperating devices in the robot cell and allows an automated planning and execution of all required tasks. This work presents the overall system architecture, the necessary subsystems and the various models in the virtual representations of the devices and products for a sufficient description of the conditions of entire robot cell to reach the before mentioned goals.

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 configure and program an entire robot cell consisting of multiple robots, tools and sensors – for example at the introduction of a new or altered product – is high and limits the utilized flexibility. 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 and shrinking lot sizes [2]. These demands have to be met with new approaches to plan the production process as well as configure and program 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. Suitable devices are selected manually and human 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 trial-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 their 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 [5–7].

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, planning, 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

* Corresponding author.
E-mail addresses: Joachim.Michniewicz@iwb.tum.de (J. Michniewicz), Gunther.Reinhart@iwb.tum.de (G. Reinhart).

http://dx.doi.org/10.1016/j.mechatronics.2015.04.012
0957-4158/

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 automatically analyzed without the need for real world testing.

To reach the before mentioned goals, a new method 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 manufacturing, which is explained in the following scenario. 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) consisting of a variety of tools, sensors, mobile and fixed robots. All these devices have a virtual representation of their skills and properties. The devices 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 all viable production processes with corresponding constrains. Processes required by the product can be automatically compared with the viable 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 fulfillment of requirements of the production process, the designer, being the client, can choose optimization criteria (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 small 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, configuration and programming 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 must be generated. The automated extraction of assembly sequences from virtual product data usually demands 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 [10–12]. An alternative approach is knowledge based reasoning. Existing assembly plans of earlier products are modified to adapt to the requirements of a new product [13–15]. Commercial computer aided design systems do not offer the extraction of assembly sequences from CAD models. Ou and Xu [16] analyzes 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 mentioned assembly sequence generators do not create data, which can be easily used to program and configure a robot. Refs. [17–19] 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. Arai et al. [20] introduces 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. Mendes et al. [21] presents 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 then 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. Naumann et al. [7] presents 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. Huckaby and Christensen [22] provides a detailed taxonomic framework, which allows modeling of skill primitives of devices of a robot cell 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 [22] 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. Keddis et al. [23] proposes 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 run-time is not possible [24–26].

Nylund et al. [27] and Salminen et al. [28] present a method, which allows the connection of different devices possessing

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

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 independent of the solution. The method is not used for automated programming and configuration of robot cells.

Lau [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. Ostgathe [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 and different feasible assembly orders from a virtual product description in order to optimally plan the required production processes as well as configure and program a modular robot cell. Abilities of the robot cell 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 method, which allows the automated and efficient production of variable products with lot size one. The approach shall enable automated programming, configuring and optimization of robot cells based on individual, solution neutral product requirements. The flexibility of modular robot cells regarding cooperating resources, reconfigurations and alternative value streams is to be utilized in order to reach an unprecedented level of adaptability. A model representing the versatility and adaptability must be introduced. Communication between devices in a modular robot cell and the products to be manufactured must be developed. Virtual descriptions of the product and the devices need to be defined. Mutual interferences and resulting constraints between devices and products must be determined and modeled adequately. Methods for automatically generating the virtual model of the product, containing different feasible process sequences, must be introduced. Furthermore a planning method allowing comparing and optimally assigning product requirements to single or cooperating devices with adequate abilities must be described. The use case, chosen to explain the method, 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)*. It contains the geometrical description of each part to be assembled, different valid assembly orders and describes the required production process solution independently. Together with the physical parts of the product, it forms the *Cyber-Physical-Product (CPP)* (see Section 6.4), a CPS according to [9].

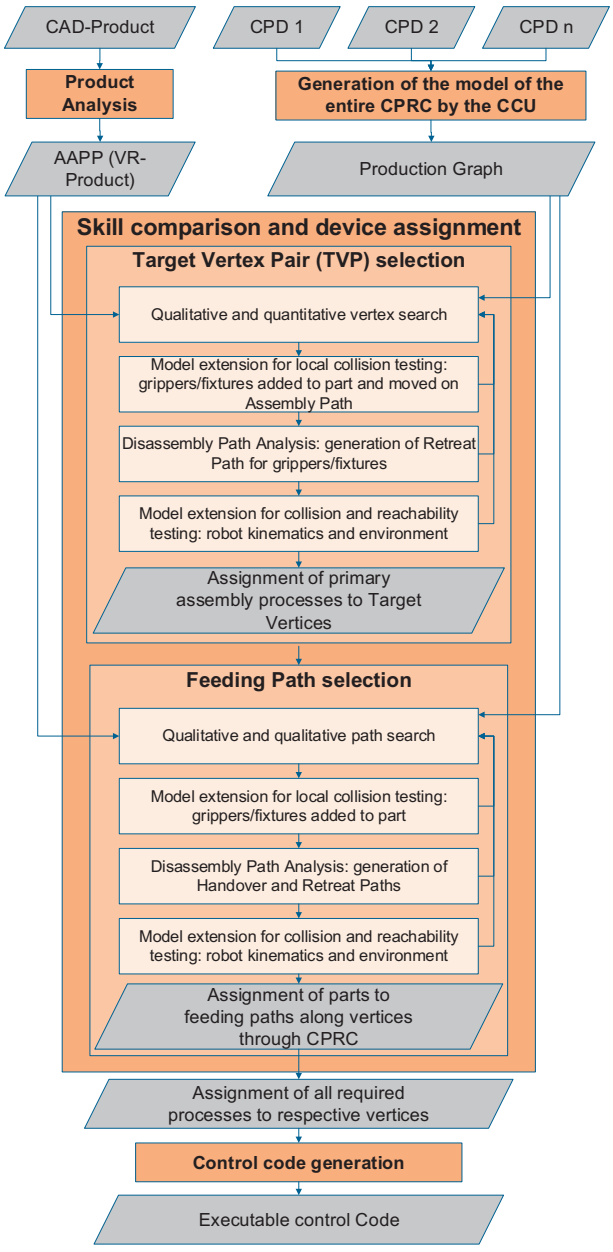

Fig. 1. Overview over all subsystems.

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)*. CPDs are CPSs [9]. A *Cyber-Physical-Robot-Cell (CPRC)* consists of a *Centralized-Control-Unit (CCU)* and any number of cooperating CPDs. The CCU is an IT-System which creates the virtual representation of the entire robot cell by detecting and analyzing all the available CPDs in it. It determines possible device configurations with corresponding abilities and constraints as well as feasible paths through the cell in each configuration. The digital model representing the entire robot cell is called *Production Graph (PG)* (see Section 6.3).

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

In a next step skills required by the AAPP and skills offered by the CPRC with the regarding constraints are compared by the CCU, allowing an evaluation of the feasibility of the production process. All assembly and feeding processes are taken into account (see Section 6.5.3).

If the CPRC meets requirements of the AAPP, the parts and processes are assigned to corresponding cooperating CPDs. If various sets of cooperating CPDs, more than one configuration of the CPRC or different assembly orders comes into question, an optimization algorithm selects the planning alternative which best match the optimization criteria defined by the user. Finally the CPRC is automatically programmed based on the information from the AAPP (see Section 6.5.4).

6.2. Cyber-Physical-Devices (CPD)

The smallest entity in a Cyber-Physical-Robot-Cell is a Cyber-Physical-Device (CPD) (e.g. a robot, a gripper, a fixture, a camera) with its individual virtual representation. The solution-independent description of the skills of each device is based on the usage of standardized Functional Primitives (FPs). Exemplary FPs in assembly are *"hold"*, *"release"*, *"move"*, *"measure force"* and *"detect pose"*. Further FPs can be added to the method in order to extend the range of describable processes.

Independent of the offered FPs, the virtual representation of every CPD contains an individual identification number, a representation of its spatial expansion and its pose in the global coordinate system of the CPRC. Changes in spatial expansion depending on the current state of the device (e.g. robot postures, open or closed gripper) are taken into account.

The further content of the virtual representation of a CPD is modular and depends on its offered FPs. Every FP has a set of standardized quantitative constraints (e.g. "hold" requires a maximum payload), whose parameters are dependent on the properties of each individual device.

Exemplary the virtual representation of devices offering the FP "move", allowing the controlled movement of the device (e.g. various types of robots), contain a simplified three dimensional workspace boundary and a detailed description of their kinematics [32]. The simplified workspace boundary represents the approximated space which the TCP of the robot could theoretically reach. For a 6-axis robot this is represented as a sphere, for a delta as a cylinder and for a SCARA as a cutout of a cylinder. The TCP of the robot is a mechanical interface other CPDs can be attached to. The detailed kinematics description contains joint types, joint limitations and link lengths for the previously mentioned robot types and is used for precise reachability tests [32–34].

Another example of Functional Primitives with a set of constraints are "hold" and "release", usually assigned to devices like fixtures, grippers and conveyors. It describes the ability to hold or grab and release other devices or parts. These have a virtual representation of their mechanical interface containing its position and orientation in the coordinate system of the device, its geometry and descriptive parameters like mass and holding force.

Every CPD has a digital description of its physical interfaces (e.g. digital, fluidic, mechanical, and electrical). This description contains type, geometry, position and orientation of the interfaces in the coordinate system of each CPD. Interfaces are divided into 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 CPDs. Also output interfaces 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 FPs and output 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. 2). The usability of its FPs also depends on the successful connection of input interfaces. An electrical gripper can only "hold" a part, if it is connected to an electrical energy source.

The FPs 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.

6.3. Production Graph – model structure of a Cyber-Physical-Robot-Cell (CPRC)

In order to utilize the entire flexibility of modular robot cells of any size, the Production Graph, a modeling approach representing a CPRCs entire solution space, has been developed. The structure of the Production Graph is presented in this chapter.

The smallest entities in a CPRC, the CPDs, can be attached to other CPDs by linking their matching physical input and output interfaces in order to gain more sophisticated functionalities. A physical combination of two or more CPDs (e.g. a gripper and a robot) is defined as a *Cyber-Physical-Device-Combination (CPDC)*, offering altered Functional Primitives. An important aspect for determining the quantitative constraints of FPs provided by a CPDC is the analysis of the mutual interaction of the corresponding CPDs based on their properties. The constraints of the separate CPDs are combined. Exemplary the maximum payload of the robot is reduced after attaching a gripper with a certain mass. Another example is the influence of the arrangement of objects (e.g. walls, conveyors, grippers attached to robots) on the accessibility and collision freedom in the CPRC. The spatial expansion models of all CPDs in the CPRC are merged into one three-dimensional model. Methods for describing the workspaces of robot systems can be found in [35–38], but had to be developed further to take the modularity of a CPRC into account.

The CCU, responsible for generating the model of the overall robot cell, represents its overall abilities and constraints in an unidirected graph model, called Production Graph (PG). The PG is composed of vertices and edges.

Vertices represent single CPDs (e.g. a fixture) or a set of CPDs able to autonomously recombine themselves into a variety of CPDCs (e.g. a robot able to link with two different grippers). The

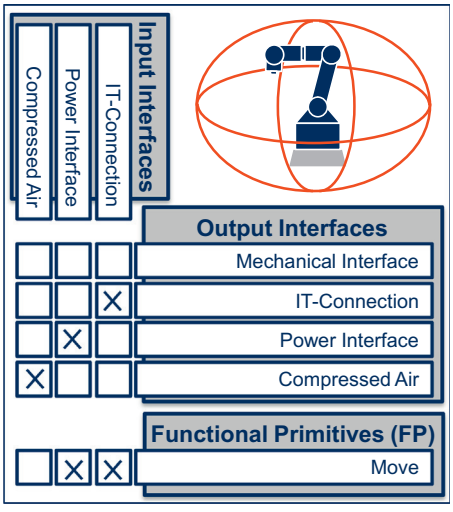

Fig. 2. Availability of output interfaces and Functional Primitives depending on input interfaces for an industrial robot.

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

requirement for a single CPD or set of CPDCs to be a vertex is to offer executable Functional Primitives required by Cyber-Physical-Products.

Edges connecting vertices in the Production Graph represent their ability to physically cooperate, for example to hand a part from one adjacent vertex to another. Two vertices are connected by an edge if their workspaces overlap.

The automated generation of the Production Graph, containing all vertices with individual sets of available Functional Primitives with corresponding constrains and edges representing the vertices ability to physically cooperate, is further described in Section 6.5.2.

6.4. Cyber-Physical-Product (CPP)

The CPP consists of the physical parts of the product to be assembled and its virtual representation, the Augmented-Assembly-Priority-Plan (AAPP). The AAPP contains information about the individual parts and the necessary assembly processes called *Tasks* (see Fig. 3). The output of a Task is a semifinished product or, if it is the last Task in the AAPP to be executed, the final product. The information from the AAPP is used to analyze the feasibility of the assembly task in a CPRC and to program the devices in the CPRC. Without knowledge about the robot cell, only primary, value adding processes [39] can be extracted from the CAD-File. Secondary processes like handling and quality control are automatically added to the AAPP after a successful assignment to suitable devices in a robot cell. The AAPP is automatically updated during the assembly process in order to synchronize the virtual model with the current state of the physical product.

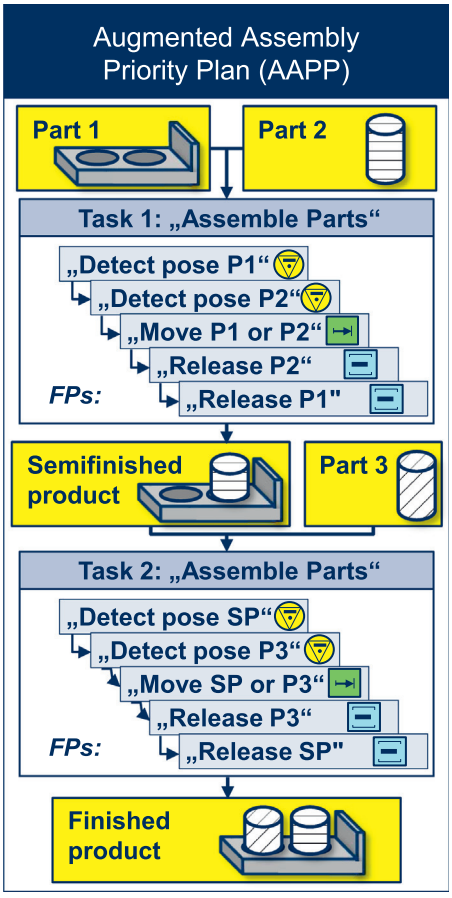

Fig. 3. The AAPP is composed of parts and Tasks. Tasks are build up of recurring sequences of Functional Primitives.

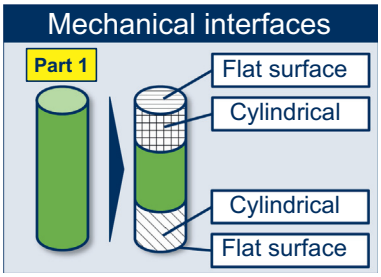

Fig. 4. Each part must have at least two mechanical interfaces. The position of the interfaces is stored in the coordinate system of the individual part.

The description of each part in the AAPP includes its geometrical properties, mechanical interfaces, mass, Final Pose in regard to other parts and feasible *Assembly Path (AP)*.

Generally each part of the product to be assembled has at least two mechanical interfaces, allowing it to be passed from one gripper or fixture to another (see Fig. 4). The description of the mechanical interface contains its position and orientation in the coordinate system of the part, its type and descriptive parameters.

The Assembly Path (AP) represents a sequence of points with orientations (poses) with a representation of the motion degrees of freedom between them, describing the necessary relative assembly movement of the parts in regard to each other. The last point in the AP is the Final Pose of the part in the fully assembled product [16].

The process requirements of the product to be assembled are described solution independently as sequences of standardized Functional Primitives (FP). Most common FPs in assembly are "hold", "release", "move", "measure force" and "detect pose". Every qualitative function a FP refers to is connected to corresponding quantitative requirements. These are dependent on the quantitative descriptions of the respective parts or intermediate products. Exemplary the FP "move" contains a desired start pose, an end pose and required degrees of freedom extracted from the Assembly Path; the FP "hold" contains constraints extracted from the parts mechanical interface description.

FPs are used for the task oriented programming of the robot cell. FPs have start and exit conditions, e.g. a FP "release" can only be executed, if a previously requested pose was successfully reached.

In order to facilitate the use of knowledge and reduce complexity, recurring sequences of FPs are connected to the previously mentioned Tasks. The most common Task in the presented system is called "assemble parts". Its goal is to assemble two parts or intermediate products. Its start condition is that both parts or intermediate products are "held" by a CPD. It consists of the following sequence of FPs: "detect pose" for both parts, "move" of either one part along the entire Assembly Path until the Final Pose is reached, "release" of both parts by the CPDs (see Fig. 3).

Furthermore the AAPP describes all feasible assembly orders by storing every possible sequence of linkages between parts and intermediate products with Tasks.

6.5. Required subsystems

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

6.5.1. Product analysis – generation of the AAPP from a CAD file

The initial input for this subsystem is a CAD file, representing the fully assembled product. It contains the geometrical

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

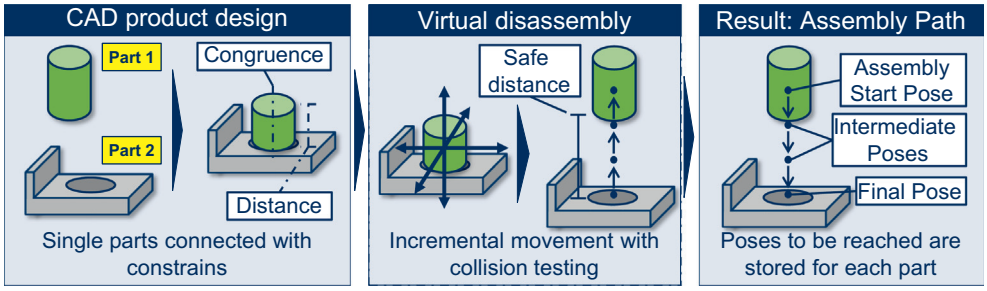

Fig. 5. Disassembly Path Analysis of a simple CAD-product.

information of every single part and the geometrical constraints between them. The output of the system is the AAPP, consisting of parts and Tasks (see Fig. 3).

Only primary production processes [39] can be extracted from the CAD file. At this stage all information is stored in the coordinate systems of the corresponding parts.

While the part geometry, mass and *Final Pose* can be read directly from the CAD file, the Assembly Path is determined by incrementally moving each part in a random direction. If a collision is detected, the movement direction is obstructed and the part is incrementally moved in another direction. If the direction is collision free, the pose is stored in the coordinate system of the part. These poses are called *Intermediate Poses*. This sequence is repeated until the part reaches a pose, in which subsequent movement in all directions is free of collisions and the distance between the parts is large enough [16]. This pose is called *Assembly Start Pose*. The Assembly Path is created by putting the determined poses in reverse order. The process described in this paragraph is called *Disassembly Path Analysis (DPA)* (see Fig. 5).

Feasible assembly orders are determined by subjecting all parts to the DPA. If a part cannot be incrementally moved from its initial pose without a collision with another part, other parts have to be disassembled first.

Due to the exponential growth of potential assembly sequences with growing part quantities, the following methods for minimizing the computational effort have been introduced. Parts contacting the least number of other parts [10], parts on the outside of the fully assembled product [40] or certain part types like screws or bolts are analyzed by the Disassembly Path Analysis first. Furthermore subassemblies, which do not have to further analyzed, can be entered manually. A limited amount of different assembly orders is generated in order to limit the solution space.

As mentioned in Section 6.4, each part has at least two mechanical interfaces (see Fig. 4). The type (cylinder, parallel surfaces, flat surface), corresponding parameters (e.g. diameter and length for a cylindrical interface) and the pose of the interface in the coordinate system of the part can be analyzed automatically or entered manually.

6.5.2. Generation of the model of the entire Cyber-Physical-Robot-Cell (CPRC)

In order to allow an automated comparison between the FPs required by the products and the FPs offered by the manufacturing system, a description of the latter needs to be generated. The following 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 CPDs in the robot cell and downloading their virtual representations, the CPRC begins the automated generation of its individual Production Graph.

Vertices can contain a single CPD or a set of CPDs able to create CPDCs. Vertices are generated automatically by analyzing positions and orientations of workspaces and interfaces of all CPDs available in the CPRC. A prerequisite is the knowledge of the positions and orientations of all CPDs in the robot cell. Vertices are created by analyzing the geometrical intersections between simplified robot workspaces (introduced in Section 6.2) and input interfaces of all available CPDs (see Fig. 6). All CPDs, whose input interfaces are inside the simplified workspace of a robot and could, together with the robot, be autonomously combined to CPDCs (e.g. a robot and a set of adjacent grippers), are assigned to one vertex. CPDs like fixtures and conveyors, which have no output interfaces allowing them to form a CPDC with other CPDs, are represented as individual vertices. The workspace of these single-CPD vertices is represented by the geometry of their mechanical interface (e.g. cuboid workspace for a vise). Based on the iteration of possible connections between the available input and output interfaces of every CPD in a vertex, all feasible CPDCs the vertex can build are determined. In a next step, the CCU can qualitatively and quantitatively analyze which full sets of FPs are available in the vertex based on the mutual interaction of the devices in each configuration. This procedure is performed for all vertices in the CPRC, allowing the determination and quantitative description of all available Functional Primitives in the robot cell (see Fig. 7).

In a second step the edges between the vertices are generated by analyzing the geometrical intersections between the vertices. The CCU analyzes intersections between simplified workspaces of robots and workspaces of other vertices like fixtures and conveyors. If an intersection is detected, the respective vertices are connected with an edge in the Production Graph, representing their ability to physically work together (see Figs. 8 and 9).

The output of the subsystem is a model containing all possible cooperation between devices in the Cyber-Physical-Robot-Cell and all Functional Primitives with corresponding constrains the CPRC can perform.

6.5.3. Comparison of Functional Primitives and device assignment

A core element of the product-centered programming and configuring of the manufacturing system is the comparison and assignment of assembly processes required by the product with those offered by the reconfigurable, cooperating devices in Cyber-Physical-Robot-Cell. The FPs provided by the vertices in the CPRC have to be optimally matched with the set of possible sequences of FPs represented in the AAPP with the aim of successfully assembling the product. In order to efficiently reduce the solution space in each sequential step of the device assignment, comparisons of requirements and restrictions excluding many alternatives and with the smallest computational effort are executed first.

In the first step, the search for the *Target Vertex Pair (TVP)* takes place. The primary production processes extracted from the CAD file, represented as *Assembly Tasks*, depict consecutive assembly operations, during which two parts or semifinished products are moved relative to each other until their Final Pose in relation to

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

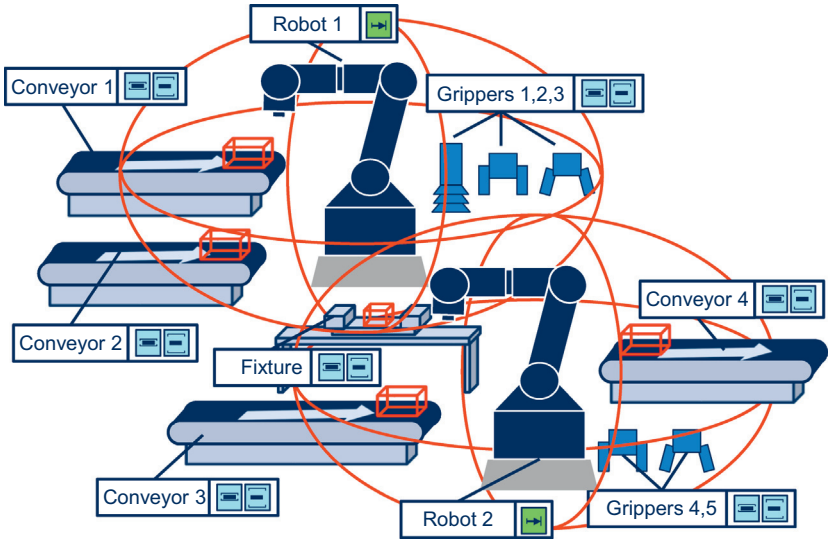

Fig. 6. Arrangement of devices in an exemplary Cyber-Physical-Robot-Cell. Workspaces of the devices are represented in red. The positions of the mechanical input interfaces of the grippers are symbolized with a dotted cross. Conveyors 1, 2 and 3 are Start Vertices (SV) for different parts, conveyor 4 is an Exit Vertex (EV) for the final product. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Vertex	CPDs in Vertex	FPs of each CPD	Simplified workspace	CPDCs available in Vertex	FPs available In Vertex
1	Conveyor 1	Hold —IF 2	Cuboid	none	Hold —IF 2
2	Conveyor 2	Hold —IF 2+3	Cuboid	none	Hold —IF 2+3
3	Conveyor 3	Hold —IF 2+3	Cuboid	none	Hold —IF 2+3
4	Robot 1	Move	Sphere		
	Gripper 1	Hold —IF 1		CPDC1: Robot1+Gripper1	Move + Hold IF 1
	Gripper 2	Hold —IF 2		CPDC2: Robot1+Gripper2	Move + Hold IF 2
	Gripper 3	Hold —IF 3		CPDC3: Robot1+Gripper3	Move + Hold IF 3
5	Robot 2	Move	Sphere		
	Gripper 4	Hold —IF 4		CPDC4: Robot2+Gripper4	Move + Hold IF 4
	Gripper 5	Hold —IF 5		CPDC5: Robot2+Gripper5	Move + Hold IF 5
6	Fixture 1	Hold —IF 5	Cuboid	none	Hold —IF 5
7	Conveyor 4	Hold —IF 5	Cuboid	none	Hold —IF 5

Fig. 7. Tabular representation of the Production Graph of the exemplary Cyber-Physical-Robot-Cell. Vertices are linked to their individual CPDs, simplifies workspaces, CPDCs and overall available Functional Primitives. The abbreviation IF stands for the interface type.

each other is reached and a required Task is finished. In order to execute the required Assembly Task, both parts must be held by a device and therefore require the FP "hold". Based on the individual quantitative requirements towards all FPs the Task is composed of (e.g. "hold" for both parts), two adjacent vertices offering all required FPs are searched in the adjacency matrix.

After assigning each part to a CPD or CPDC, the geometrical model of both parts to be assembled in their Final Poses is extended at their mechanical interfaces only by the geometrical models of the previously chosen CPD offering a suitable FP "hold", usually a gripper or a fixture (see Fig. 10a). In a next step, the required movement of the parts merged with the geometrical gripper models along the Assembly Path is simulated in order to detect collisions (see Fig. 10b). If a collision is detected, another mechanical interface of the part or another type of available gripper is tested in order to find a suitable solution.

If the Assembly Path can be completed free of collisions, the *Retreat Path* of the CPDs offering the FP "hold" away from the semifinished product after the successful assembly needs to be determined. This is realized by loading the geometrical model of the open gripper in its Retreat Start Pose in regard to the semifinished product and performing the Disassembly Path Analysis (see Section 6.5.1) in order to determine a Final Pose which is collision free in all directions and in safe distance to the semifinished product (see Fig. 10c).

After determining a suitable pair of CPDs offering the FP "hold" and analyzing the feasibility of the Assembly and the Retreat Paths, the model is furthermore extended. The reachability and collision

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

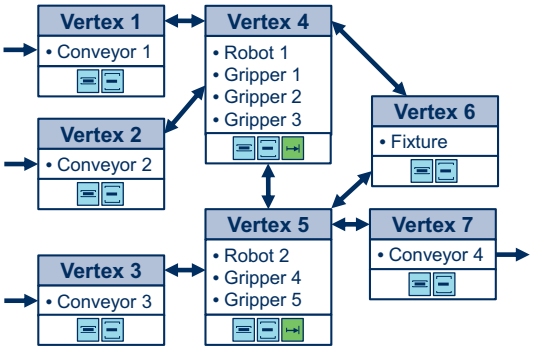

Fig. 8. Graphical visualization of the Production Graph of the exemplary Cyber-Physical-Robot-Cell. Vertices are shown with their corresponding CPDs as well as symbols of the available Functional Primitives.

	V1	V2	V3	V4	V5	V6	V7
V1	0	0	0	1	0	0	0
V2	0	0	0	1	0	0	0
V3	0	0	0	0	1	0	0
V4	1	1	0	0	1	1	0
V5	0	0	1	1	0	1	1
V6	0	0	0	1	1	0	0
V7	0	0	0	0	1	0	0

Fig. 9. Representation of the Production Graph as an adjacency matrix.

freedom of all previously determined paths is tested, taking the detailed kinematics of the robots in the selected configuration and the spatial expansion of all elements (e.g. all CPDs, static obstacles, parts to be assembled) in the robot cell into account. If all required points can be reached and no collisions are detected, the linkage between the FPs offered by the CPRC and all FPs required by the CPP, allowing the execution of the primary production processes [39], is complete. If no feasible pair of vertices can be found, another assembly order from the AAPP can be taken into account.

The next step is selecting the *Feeding Path* of the parts to be assembled along vertices through the CPRC. The parts enter the CPRC trough conveyors or feeding systems, which are represented as vertices and distinctively named *Start Vertices (SV)*. Finally the finished product must be handed to an *Exit Vertex (EV)*, allowing the product to exit the CPRC (see Fig. 6). In order to execute the required Assembly Tasks at corresponding Target Vertex Pairs, feasible feeding paths for all required parts through the CPRC from the Start Vertex to the Target Vertex and further to the Exit Vertex must be found. The paths of the parts required for their feeding (secondary production processes) cant be extracted from the CAD file of the fully assembled product and are generated in the following steps.

Initially a part is assigned to a SV. If the exact pose of a part in a vertex is unknown, it can be determined with adequate sensors, e.g. with a camera system. One of the parts at least two mechanical interfaces (see Section 6.4) is occupied by the SV. In a next step the CCU analyzes qualitatively and quantitatively, if CPDs belonging to the adjacent vertices can provide the free mechanical interface of the part to be fed. The CCU tries to find the shortest possible connection from the Start Vertex of a part to its Target Vertex in the adjacency matrix utilizing an altered form of the Dijkstra algorithm [41], which takes the constraint that two adjacent vertices cannot use the same mechanical interface of the part to be fed into account.

The next step is the determination of accurate *Handover Paths* and Retreat Paths for each part exchange between vertices. A Handover Path describes the handling of a part from one CPD to another. The Retreat Path describes a collision free retreat movement of the CPDs after the handover. First, the geometrical part model is, at its mechanical interfaces, extended by the geometrical models of both CPDs offering the FP "hold" from the two adjacent vertices (see Fig. 11a). By applying the Disassembly Path Analysis (see Fig. 11b and d) to both CPDs, collision free Handover and Retreat Paths in relation to the coordinate system of the part for both CPDs can be determined (see Fig. 11c and e). If a collision (e.g. between the CPDs) is detected, another CPD able to hold the part from the selected vertices or another sequence of adjacent vertices through the Production Graph must be chosen.

Feasible Handover and Retreat Paths are tested, similar to the Target Vertex Pair analysis, in regard to reachability and collision freedom by extending the model by detailed robot kinematics and the spatial expansion of all elements in the robot cell. This procedure is repeated for each part or semifinished product and every pair of adjacent vertices exchanging the parts between them to analyze the feasibility of the Feeding Path of the parts along the vertices trough the entire robot cell.

If all required Assembly, Handover and Retreat Paths passed all above mentioned tests, the AAPP, which initially only contained primary processes, is extended by the necessary secondary processes like feeding operations. All FP in the AAPP are unambiguously linked to respective CPDs and CPDCs.

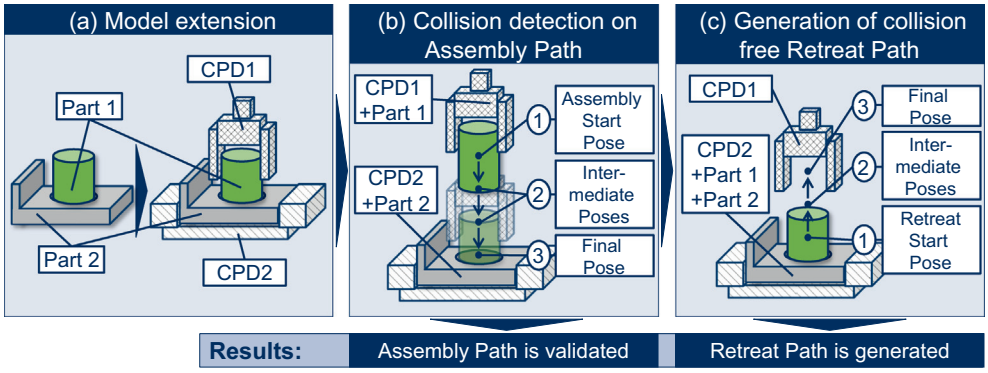

Fig. 10. Determination of a suitable pair of CPDs for holding the parts to be assembled.

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

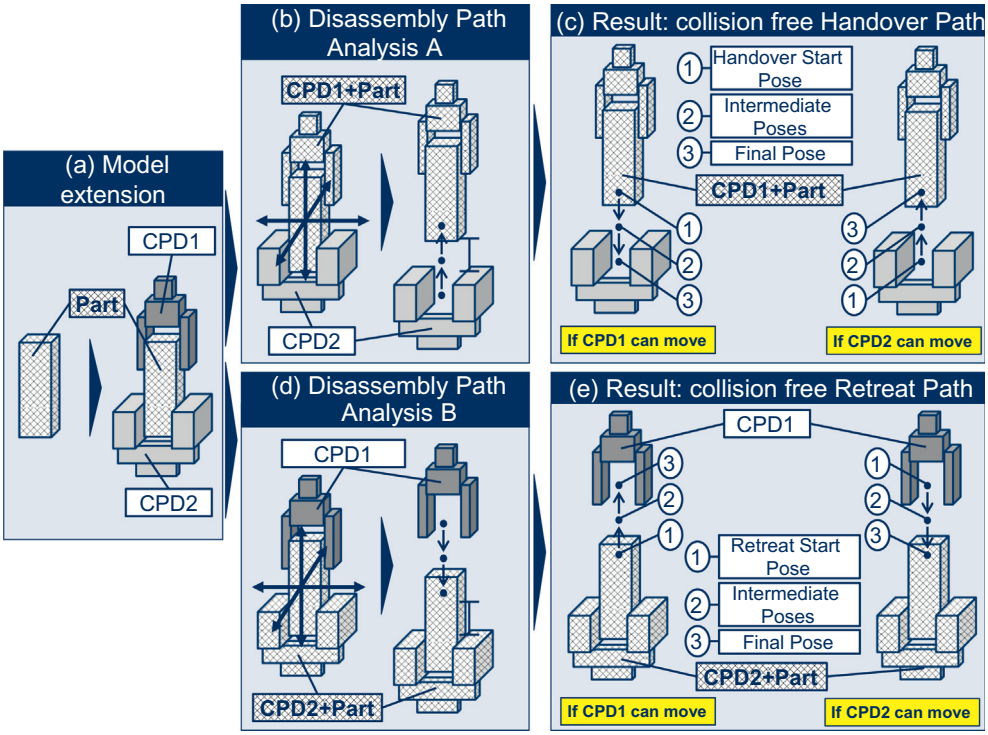

Fig. 11. Determination of feasible CPDs for feeding parts between vertices. The part is handed from CPD1 to CPD2. Handover and Retreat Paths are generated. The utilization of the Paths is dependent on which of the two CPDs can change its position.

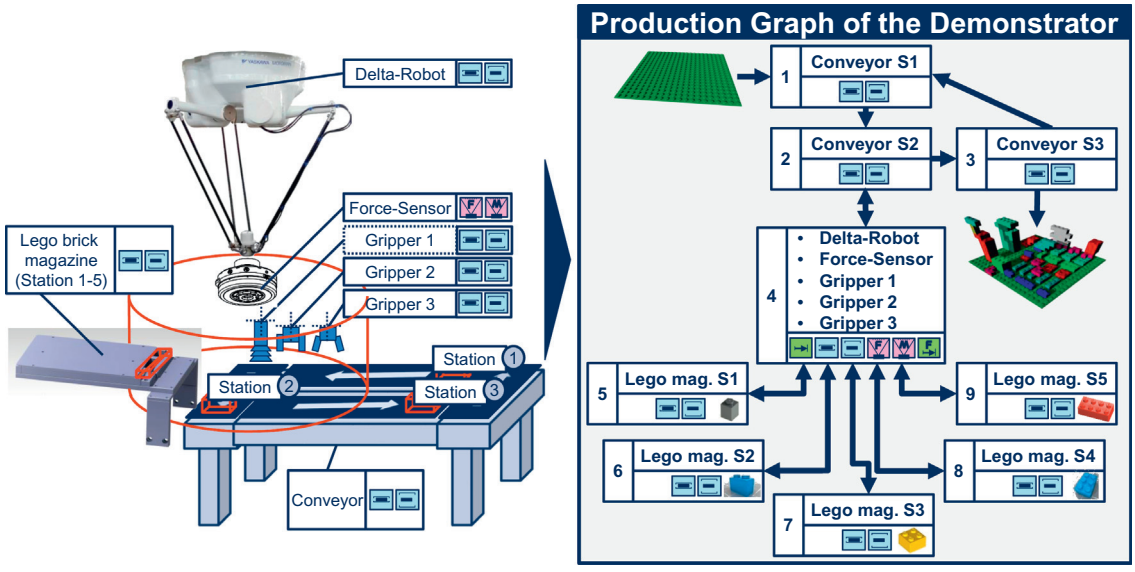

Fig. 12. Representation of the Cyber Physical Devices in the demonstrator and its automatically generated Production Graph. Due to the fact that the conveyor can only move in one direction, its vertices, representing the three stations of the conveyor, are connected by unidirectional edges.

If information which is necessary for the comparison or programming is not available, e.g. a quantitative test of the Functional Primitive "hold" is supposed to be executed and the mass of a part is unknown, the system autonomously queries the user, e.g. the product designer, and calls for the required input.

Due to the fact that the AAPP is created from a CAD file, the comparison between Functional Primitives offered by a Cyber-Physical-Robot-Cell and those required by a product yet only existing in a digital description is possible. Product requirements which cant be met by the current production system are determined automatically. This allows an informed decision in regard to changing corresponding product details or adding CPDs to the CPRC, which satisfy the induced product requirements. The presented functionality highly reduces the number of iterations between product designers and process planners, allowing a faster integration of new products or resources.

6.5.4. Optimization and task-oriented programming

The optimization criteria taken into account while choosing a feasible Target Vertex Pair and feeding paths through the CPRC are a lowest possible number of handling operations between vertices and a minimal amount of necessary reconfiguration during

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012

the entire production process of a product. Further optimization criteria like production time or production costs could be integrated into the system. In a scenario with highly flexible robot cell and a variety of products to assemble, production planning and control is taken to a new level of flexibility and transparency.

The programming of the CPRC can be initiated, as soon as all necessary variables in the AAPP are known and the optimal assignment between required and provided FPs in the CPRC took place. The information from the Functional Primitives from the AAPP is send to the Functional Primitives of the CPDs in order to generate the control code. The generated assembly and feeding commands are executed sequentially based on the current state of the affected devices, the positions of the parts and the state of the product. A solution-independent, task oriented programming takes place.

7. Demonstrator

It order to demonstrate the potentials of the presented approach, a real life demonstrator was designed and build. The selected use case, due to its flexibility and the huge amount of alternative process sequences, is the assembly of Lego bricks. All devices in the robot cell have an individual virtual representation and consequently are CPDs. The robot cell contains a delta robot with an automated tooling system, allowing an automated selection between three different grippers. Each of the grippers is suitable for certain Lego bricks. A conveyor allows the feeding of square Lego baseplates. The different Lego bricks are fed from a magazine in the robot workspace.

A specially developed interface allows the convenient design of CAD-models of a Lego products, which are analyzed with the methods from Section 6.5.1. The software automatically analyzes feasible assembly orders and stores them in the products AAPP. The Production Graph of the robot cell is automatically generated based on the CPDs in the cell (6.5.2) (see Fig. 12) and the feasibility of the assembly task is analyzed (6.5.3). Bricks in the virtual model which cant be assembled, e.g. due to collisions or insufficient parameters of the FP "hold" offered by the available grippers, are detected and highlighted in terms of color on the graphical interface, allowing the user a convenient manual redesign based on this information. If all processes required be the product are optimally assigned to suitable, available CPDs or CPDCs, the task oriented assembly begins.

8. Conclusion

The presented method shows the capabilities of modular robot systems equipped with Cyber-Physical-Devices and -Products. 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, optimally planning the assembly and programming the robot cell allows a variety of new benefits. Due to the analytical description of the constraints of the Functional Primitives of the robot cell the feasibility of the assembly process can be determined with certainty. Necessary changes to the product or the robot cell are formulated precisely and transparently by automatically determining product requirements exceeding the abilities available in the robot cell, accelerating the introduction of new products or new devices. In order to meet the requirements of each individual product to be assembled, the robot system is configured and programmed autonomously. The planning and control algorithm works on a new level of detail: it can take all available configurations of the production system and all possible production sequences of the products into account, greatly increasing its flexibility.

The overall system is successfully validated by linking it to simulation software and a prototypical demonstrator. The simulation allows testing the feasibility of the approach for robot cells of any size whereas the demonstrator proves the operability in connection with stock robot systems and programmable logic controllers.

Acknowledgements

The presented work was conducted as part of the research project CyProS [42], funded by the German Federal Ministry of Education and Research.

References

[1] Reinhart G, Krug S. Robotersysteme in der kleinserie – effizienz von der planung bis zum einsatz. iwb seminarberichte 91, Mnchen, Utz; 2009. p. 1–16.
[2] Abele E, Reinhart G. Zukunft der produktion. Mnchen: Carl-Hanser-Verlag; 2011. ISBN: 3446425950.
[3] Wiendahl H-P, ElMaraghy HA, Nyhuis P, Zh MF, Wiendahl HH, Duffie N, et al. Changeable manufacturing classification, design and operation. CIRP Ann-Manuf Technol: Ann CIRP 2007;56(2).
[4] Anderl R, Eigner M, Sendler U, Stark R. Smart Engineering – Interdisziplinre Produktentstehung. acatech Diskussion April 2012, acatech Deutsche Akademie der Technikwissenschaften, Herausgeber Prof. Dr.-Ing. Reiner Anderl Technische Universitt Darmstadt.
[5] Krug S. Automatische konguration von robotersystemen. Herbert Utz Verlag GmbH; 2013. ISBN: 978-3-8316-4243-4.
[6] Reinhart G, Httner S, Krug. Automatic configuration of robot systems – upward and downward integration. In: Reinhart G et al., editors. (Hrsg.): proceedings of the 4th international conference on intelligent robotics and applications (ICIRA 2011). Aachen, 06.-08.12.2011. Heidelberg: Springer; 2011. p. 102–11.
[7] Naumann M, Wegener K, Schraft RD. Control architecture for robot cells to enable PlugnProduce. In: 2007 IEEE international conference on robotics and automation Roma, Italy; 10–14 April, 2007.
[8] Geisberger E, Broy M. Integrierte forschungsagenda cyber-physical systems. Agenda CPS acatech Studie March 2012. Dr. Eva Geisberger fortiss GmbH Guerickestrae 25 80805 Mnchen.
[9] Reinhart G, Engelhardt P, Geiger F, Philipp TR, Wahlster W, Zhlke D, et al. Cyber-physische produktionssysteme. Werkstattstechnik online 103 Jahrgang. H. 2 Copyright Springer-VDI-Verlag GmbH & Co. KG, Dsseldorf; 2013.
[10] Gu P, Yan X. CAD-directed automatic assembly sequence planning. Int J Prod Res 1995;33(11):3069–100.
[11] Bley H, Bossmann M. Standardisierte produktmodelle fr die automatisierte montageplanung featurebasierte montageplanung untersttzt den simultaneous-engineering-prozess. wt Werkstattstechnik online Jahrgang 2005;95:H.9.
[12] Rhrdanz F, Mosemann H, Wahl F. Generating and evaluating stable assembly sequences. Adv Robot 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):252–66.
[14] Yin ZP, Ding H, Li HX, Xiong YL. A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Comput Aided Des 2003;35(1):37–56.
[15] Fan J, Dong. KVAS – a knowledge-based virtual assembly system. In: Proceedings of the IEEE international conference on systems, man, and cybernetics, Taipei, Taiwan, October 2001; 2001. p. 1041–6.
[16] Ou Li-Ming, Xu Xun. Relationship matrix based automatic assembly sequence generation from a CAD model. Comput-Aided Des 2013;45:1053–67.
[17] Mosemann H, Wahl FM. Automatic decomposition of planned assembly sequences into skill primitives. IEEE Trans Robots Autom 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. In: Proceedings. 2001 IEEE/RSJ international conference on, vol. 3. p. 1458–64.
[19] Mosemann H. Beitrge zur Planung, Dekomposition und ausfhrung von automatisch generierten roboteraufgaben. Dissertation, Shaker Verlag, Aachen; 2000. ISBN: 3-8265-7710-8.
[20] Arai T, Aiyama Y, Maeda Y, Sugi M, Ota J. Agile assembly system by "Plug and Produce". Ann CIRP 2000;49(7) [received 03.01.2000].
[21] Mendes JM, Leito P, Colombo AW, Restivo F. High-level Petri nets for the process description and control in service-oriented manufacturing systems. Int J Prod Res 2012;50(6):1650–65.
[22] Huckaby J, Christensen HI. A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics. AAAI workshop – technical report, 2012 AAAI workshop; Toronto, ON, volume WS WS-12-06 2012. p. 94–101. ISBN: 978-157735571-7.
[23] Keddis N, Kainz G, Buckl C, Knoll A. Towards adaptable manufacturing systems. In: Proceedings of the 2013 IEEE international conference on industrial technology (ICIT 2013); February 2013.
[24] Bengel M. Workpiece-centered approach to reconfiguration in manufacturing engineering. Heimsheim: Jost-Jetter Verlag; 2010. ISBN: 3-939890-60-X.
[25] Malec J, Nilsson A, Nilsson K, Nowaczyk S. Knowledge-based reconfiguration of automation systems. In: Proceedings of the 3rd annual IEEE conference on

automation science and engineering. Scottsdale, AZ, USA; September 22–25, 2007.
[26] Malec J, Nilsson K, Bruyninckx H. Describing assembly tasks in declarative way. In: Proc ICRA 2013 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. In: 2008 manufacturing systems and technologies for the New Frontier. The 41st CIRP conference on manufacturing systems, Tokyo, Japan; May 26–28, 2008. p. 103–6.
[28] NylundSalminen K, Nylund H, Andersson P. Role based self-adaptation of a robot DiMS based on system intelligence approach. Flexible Autom Intell Manuf (FAIM) 2009.
[29] Lau J. Methodik fr eine selbstoptimierende produktionssteuerung. Mnchen: Herbert Utz Verlag; 2010. ISBN: 978-3-8316-4012-6.
[30] VDI Richtlinie 2860. Montage- und handhabungstechnik; handhabungsfunktionen, handhabungseinrichtungen; begriffe, definitionen, symbole. Dsseldorf: VDI-Verlag; 1990.
[31] Ostgathe M. System zur produktbasierten steuerung von Ablufen in der auftragsbezogenen fertigung und montage. Herbert Utz Verlag Mnchen; 2012. ISBN: 978-3-8316-4206-9.
[32] Siziliano B, Sciavicco L, Villani L, Oriolo G. Robotics – modelling, planning and control. Springer Verlag London Limited; 2010. ISBN: 978-1-84628-641-4.
[33] Craig J. Introduction to robotics – mechanics and control. 3rd ed. Pearson Education Inc.; 2005. p. 101–29. ISBN: 0-13-123629-6.
[34] Zsombor-Murray PJ. Descriptive geometric kinematic analysis of clavels "Delta" robot. USA: Centre of Intelligent Machines, McGill University; 2004.
[35] Goyal K, Sethi D. An analytical method to find workspace of a robotic manipulator. J Mech Eng 2010;ME 41(1):25–30.
[36] Alicatore D. Determining manipulator workspace boundaries using the Monte Carlo method and least squares segmentation. In: Proceedings of 23rd ASME mechanisms conference; August 1994.
[37] Miller K, Petitt JD. Six-dimensional visualisation of end-effector pose using colour spaces. In: Proc 2002 Australasian conference on robotics and automation (ACRA2002); November 2002. p. 216–21.
[38] Zacharias F. Knowledge representations for planning manipulation tasks. Technische Universitt Mnchen, Diss.; 2011. ISBN: 978-3-642-25182-5.
[39] Lot ter B. Montage in der industriellen produktion: ein handbuch fr die praxis. Berlin: Springer; 2006. ISBN: 978-3-540-21413-7.
[40] Mok S, Ong K, Wu C. Automatic generation of assembly instructions using STEP. In: Proc international conference on robotics & automation; 2001. p. 313–8.
[41] Cormen T, Leiserson C, Rivest R, Stein C. Section 24.3: Dijkstra's algorithm. In: Introduction to algorithms. MIT Press and McGrawHill; 2001. p. 595–601. ISBN: 0-262-03293-7.
[42] www.projekt-cypros.de.

Please cite this article in press as: Michniewicz J, Reinhart G. Cyber-Physical-Robotics – Modelling of modular robot cells for automated planning and execution of assembly tasks. Mechatronics (2015), http://dx.doi.org/10.1016/j.mechatronics.2015.04.012