An Approach for Configuration of the Industry 4.0 Technologies on Production Systems

Daning Wang, Christoph Knieke, Helge Fischer, Andreas Rausch
Technische Universität Clausthal, Institute for Software and Systems Engineering
Arnold-Sommerfeld-Straße 1, 38678 Clausthal-Zellerfeld, Germany
Email: {daning.wang|christoph.knieke|helge.fischer|andreas.rausch}@tu-clausthal.de

***Abstract*—From embedded systems to intelligent embedded systems and Cyber-physical Systems (CPSs), the production system is always evolved with the challenge of rapid technology change. But the redesign and development of a complex production system is considered a hard task and with high risk. This paper provides an approach for the managed evolution of a complex production system, which is understood as a CPS and presents a unified view of computing systems that interact strongly with their physical environment. This approach is used to guarantee the consistency between the system evolution requirements and system implementation during the evolution of this production system, which is driven by the using of Industry 4.0 (I4.0) technologies. Furthermore, the cost of implementation can be optimized with this approach. At the end of this paper, two cases are used to evaluate this approach to ascertain the suitability for the managed evolution of production systems.**

Keywords–Architecture Evolution; Industry 4.0; Configuration of Components; Cyber-physical System; Production System.

I. Introduction

The digital manufacturing and smart factory are two important application areas of I4.0 technologies, which are synonymous with highly flexible production. I4.0 technologies enable companies to offer highly individualized products by linking the internet to conventional processes and services, and to actively involve their customers very early in the development process [1]. The CPS plays an important role in the areas of digital manufacturing and smart manufacturing, where it combines the physical part with the cyber part in a holistic way. The two parts have to flexibly and dependably adapt to each other to adapt to the changing system environment.

A production system is a classical CPS, which consists of the physical part like assembly stations, warehouses, transport belts, etc. and the cyber part like the control programs and the software protocols, etc. These are connected together as an integrated complex production system. In general, a production system is not defined perfectly at the beginning and should permanently be operated in order to raise the productivity or meet changing requirements [2].

In this paper, an approach is introduced to generate a set of configuration plans for the implementation of the evolved production system according to the introduction of the I4.0 technologies on the ongoing production system. In order to describe this managed evolution of the production system, the ongoing production system is modeled with a component oriented modeling language, where the components in this production system are connected together as an integrated model and input model for the approach. This model is equivalently transformed to a graph representation, which keeps the system structure and properties of the components in the model [3]. This graph generates a set of different graphs by using of graph-based algorithms, where each generated graph represents a configuration plan of the new production system. By applying user defined combination rules, the configuration plans, which can not meet the requirements of the new production system, are detected and canceled. The rest of the generated configuration plans meet the defined combination rules and requirements in the new production system. The ones that are cost-optimal will then be simulated and implemented as a new production system. This implemented configuration plan can be continually evaluated into the second iteration of system evolution.

The paper is organized as follows: Section II gives an overview on the related work in the field of production system evolution. The system requirements, restructure of the input models, and the implementation of this approach are introduced in Section III. Two application cases are introduced in Section IV to evaluate the efficiency of the approach. Finally, Section V concludes.

II. Related Work

I4.0 is the short name for the fourth industrial evolution. The technologies of I4.0 can improve the quality and competitiveness of products, but there are few opportunities for the Small and Medium-size Enterprises (SMEs) to participate and take advantages of this trend. One major challenge is the lack of IT specialists to develop technical innovations. For example, recent studies in Germany [4] [5] show that three-quarters of the SMEs are unable to find the proper experts to bring IT innovations and the digital transformation forward. Another associated issue is also the difficulty to gather specific information which they need to adopt I4.0 technologies and solutions.

An information portal provides access to the research results developed by Stechert and Franke [6], where the basic approaches for digitization were revealed. These approaches were used to help the digitization of the product development, which was driven by the functional areas of the I4.0 technologies. However, the applications of concrete I4.0 technologies were not introduced in this study.

In the project "Intro 4.0" [7], the specific I4.0 solutions were developed and introduced to the participating industrial enterprises. The findings of the implementation of these solutions were used to derive the recommendations of these I4.0 solutions to more industrial partners. However, a comparison of the alternative I4.0 solutions was not considered.

In the work of Simko et al. [8], a CPS specific modeling language (CyPhyML) was developed and introduced to model the structure and behaviour of physical and cyber components in a CPS. The CyPhyML supported not only the non-causal modeling, but also the causal modeling in a hierarchical composition. The authors formalized the CyPhyML model with a tuple structure, which comprises sets of components by different types, sets of ports, sets of containment functions for

 ISBN: 978-1-61208-771-9

design elements and component assemblies, and sets of flows by different types. An important advantage of the CyPhyML was that the structural and behavioural specifications of a CPS can be written in one model, whereby both can be used for deductive reasoning.

Blochwitz et al. [9] developed a standardized interface named Functional Mock-up Interface (FMI) and introduced it in their work. This FMI is based on the framework of the MODELISAR project and was used for the coupling of various simulation modules in MODELISAR. That made it possible to integrate the different simulation modules together with the common interfaces. In the work of Blochwitz et al., a master simulation is introduced to couple the appropriate different modules together. But the data exchange between the different modules is not supported.

To summarize, none of the approaches provides a suitable configuration plan for implementation by correlating of the concrete I4.0 technologies and solutions during the evolution of production system. Thus, in this paper we introduce an approach which generates the configuration plan of a production system integrating the corresponding I4.0 technologies and solutions. That enables the evaluation of the proposed integration before to implement a new production system.

III. Approach

The approach has to be realized within the scope of a suitable environment and a clear implementation process, which will be introduced in this section. At first, the system requirements of this application are briefly explained. Subsequently, the restructuring of the input models for this application is introduced. Then, the implementation of this approach is introduced by using a class diagram. The necessary mathematical basics and fundamentals of this approach were already introduced and exampled in a previous paper [10].

A. System requirements

Our application is named “Solution generation system for the managed evolution of a production system”. The application contains a Graphical User Interface (GUI) and a part called “Generating”. The system environment consists of an *industry 4.0 technologies expert*, and a *production system planner*. The expert offers a set of existing I4.0 technologies for the managed evolution of the production system by using the GUI after his professional analysis on the ongoing production system. The production system designer models the ongoing production system and gives this model as an input model into the application by using the GUI. Meanwhile, he/she has to define the configuration rules, which describe the allowed configurations between the components in this production system. By using this application, the production system designer gets a set of alternative models as the solutions of the managed evolution of this production system, which is visualized by the GUI. The algorithms and functions of the approach are implemented in the “Generating” part.

B. Restructure of the input models

In practice, a production system is typically described and analyzed by using different models, where each model focuses on a fixed set of concerns on the system. That enables the system planners and engineers to understand a production system from different disciplines. The input model describing the ongoing production system will be reformed with a key-value data structure in a pair of documents, which serves as the basis for the later data processing. One document describes every component in the production system, and the other one represents the connection relationships between the components. In addition, the I4.0 technologies offered by the expert have to be described with the same data structure as the components in the production system. The configuration rules are reformed to a two tuple structure, which represents the configuration relationships from one object to another object. Besides, the targeted production system models have the same representation data structure as the ongoing production system.

C. Implementation

A class diagram is used to describe the system structure reflecting the functional requirements of the application and represents the organization and arrangement of interrelated components in a system. The class diagram in Figure 1 shows the system structure of this application, which is implemented by the object oriented programming language Java. The class `home` implements the graphical user interface for the I4.0 technologies expert and the production system designer. It provides the generic organizing and structuring of this application and the application starts with the main function in this class. Class `Algorithms` is an algorithms library comprising all of the algorithms in this application like the algorithms for path morphism, model transformation to graph structure, etc., which are called by the class `home` to implement the functional requirements in this application. Class `SystemRules` provides the combination rules for the class `Algorithms` and exchanges the information with the GUI. The transformed input model is stored in classes `node`, `ibdNode`, and `graph` providing the graph structure to keep the descriptions of the components and connection relationships in the production system.

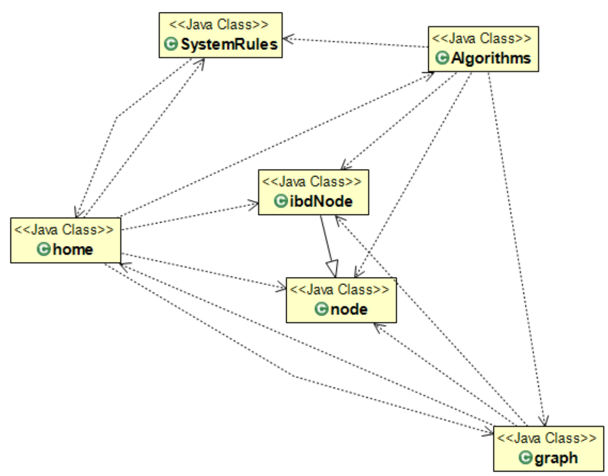

Figure 1. Class diagram of the application

IV. Case studies

In this section, two application cases are introduced to evaluate the efficiency of the approach. On the one hand, the development risks during the managed evolution of a production system should be reduced by using this approach. On the other hand, the reconstruction costs of the implementation for the targeted production system should be optimized. Therefore, the evaluation can be divided into two parts: the development risks evaluation and the economic evaluation. One case in this section is a laboratory model of a conveyor system with *Automated Storage and Retrieval System* (ASRS). The other one is part of a project named: *Methods and tools for the synergetic conception and evaluation of Industry 4.0 solutions*, in short “Synus”.

A. Case 1: Conveyor System with ASRS

This laboratory model of a conveyor system with ASRS is defined as an ongoing production system and modeled with Internal Block Diagram (IBD), which provides the internal view of a system block and represents the assembly of all blocks within the main system block. The composite blocks are connected to each other through ports/interfaces and connectors. In this case, the mechanical part in this system comprises four conveyor belts in a cycle form, a buffer belt, a RFID read/write sensor, a photoelectric sensor, a warehouse and a gripper robot. The automated tasks in this system are controlled through two industrial programmable logic controllers (PLC) of Siemens. The mechanical part and the automated part are connect to each other over the Ethernet to ensure the safety and reliability of the connection. A computer is used as a human-machine-interface (HMI) to exchange the information between workers and PLCs. In this conveyor system, wares like machine parts should be transported with the conveyor system from warehouse to the hall for the painting and dry processes by using the gripper robot, conveyor belts and the buffer-belt in accordance with the production plan. One worker (the Worker 2 in Figure 2) defines the color information of the ware by using the computer and the RFID read/write sensor, when any ware arrives at the RFID sensor. The wares will continue to be transported to a painting and dry hall. After the painting and dry processes, the wares will be transported back to the warehouse by using the buffer belt and conveyor belts and wait for the following manufacturing processes. The storage of painted wares must follow certain rules and standards, e.g., the wares with different types, sizes, materials or paint colors can be divided into different groups and stored in the designated location or floor. For this reason, a worker (the Worker 1 in Figure 2) stands by the buffer belt and sorts the wares by a predefined sorting order. This is a manual task.

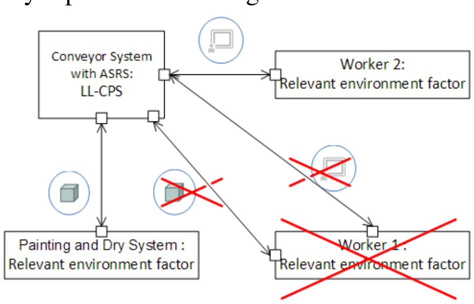

Figure 2. Comparison of the system environments between the ongoing system and the targeted system

From the perspective of production efficiency, this ongoing system is not perfect, because the manual work of Worker 1 in this ongoing system could cause an increase in production time. An automated machine definitely would have higher production efficiency. Furthermore, the worker who stands by the buffer belt repeatedly performs the same task (sorting the wares), which increases the risk of making mistakes. In many factories, this is a main reason for the poor product quality.

Hence, a new system as a targeted system is clearly defined. By using the approach, a set of solutions are generated. Therein two solutions are implemented and used to evaluate the efficiency of the approach. The first solution is named “solution 1”. There is no worker (Worker 1 in Figure 2) standing by the buffer belt to sort the wares that come back from the painting hall in solution 1. Instead of the worker, a new RFID read sensor is procured and installed on the conveyor belt (in Figure 3). It is used to read the color information from the wares. Simultaneously, this new sensor will also replace the account work of the wares of the photoelectric sensor (`PH: Sensor` in Figure 3). Not only the physical components, but also the software code in the control system and information system has to be changed to adapt to the reconstruction in the system environment and mechanical system.

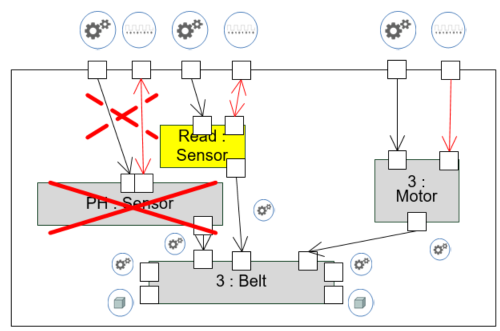

Figure 3. Comparison of the mechanical components between the ongoing system and solution 1

Figure 4 shows the changes of the software code in the control system of the targeted system compared with the ongoing system. In this figure, symbol “-” marks the deleted code parts during the managed evolution of this ongoing system. The symbol “+” marks the new added code parts and the “△” labels the code parts that changed the executing place.

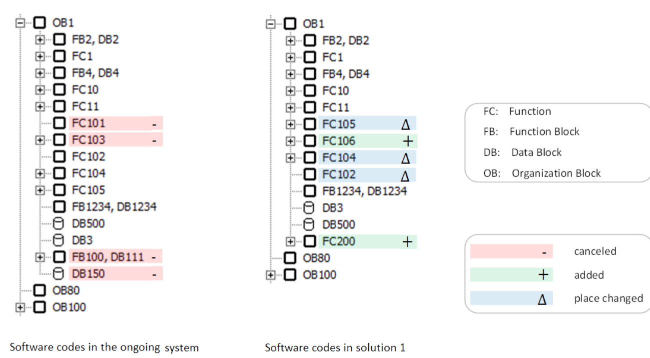

Figure 4. Comparison of software code in the PLC control system between the ongoing system and solution 1

The second solution is solution 2, where the system environments have the same changes as the solution 1 (see Figure 2). But the components in mechanical system have no changes compared with the ongoing system. In order to reach the requirement in the targeted system, the existing RFID read/write sensor is used to write the color information into the ware, when the ware reaches it for the first time, and it is reused to read the color information from the ware, when the ware reaches it again. In this situation, the other mechanical components and software code have to be adapted by using the controllers to reach this task. Accordingly, all four conveyor

 ISBN: 978-1-61208-771-9

belts have to continually transport the wares in a cycle after the painting color process to enable the wares to reach the existing RFID read/write sensor again. Meanwhile, the gripper and the photoelectric sensor are blocked to let the ware run in a cycle and activates again, when the read/write sensor obtains the full information of all wares.

In this case, the matching of functions is identified as the most import risk factor for the development risk evaluation of the managed evolution of production systems. The functional requirements are specified with the following points: (1) There is no worker standing by the buffer belt to sort the wares. (2) The color information is read by using the RFID sensor. (3) The wares are retrieved through the gripper robot with the sort information in the predefined floor in the warehouse. In order to evaluate these two solutions, we have implemented these two configuration plans as two production systems and evaluate their development risks with the specified functional requirements. The solution 1 and 2 satisfied all of the functional requirements.

For the evaluation of the economic efficiency, the direct costs are defined as the exclusive costs in the total reconstruction. That means, the indirect costs, the non-construction related costs, the time dependent costs, the software code rewriting costs, e.g., are not included in the total reconstruction costs. The reconstruction costs for different components are specified by characters. The addition of a new hardware component is among the most expensive in all of the reconstruction actions. The modifications of hardware and software components incur more costs than their deletions. After the evaluation, the solution 2 is confirmed as the optimal solution in the set of all solutions, which were generated by using the approach.

B. Case 2: Project Synus

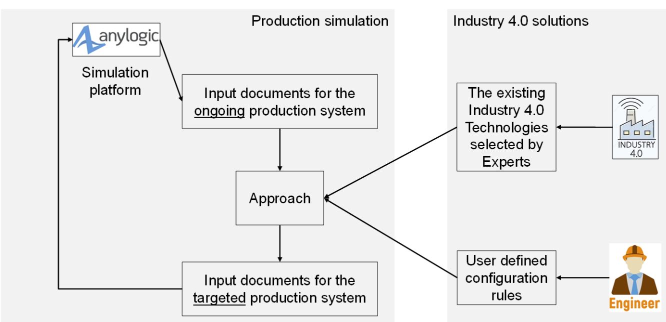

Figure 5. System architecture in project Synus

Figure 5 shows a concept for the system architecture in project "Synus" [11]. The evaluating factors like production time, energy consumption and processing costs are extracted for the evaluation of the ongoing production system. During the analysis of the result of the evaluation by the experts, some current I4.0 technologies are proposed to improve the production performance of the ongoing production system. The application of these technologies drives the evolution of this ongoing system. The ongoing system is simulated by using a simulation software "AnyLogic" [12]. Our approach generates a set of configurations of the components in the targeted status of this production system, which have to meet the configuration rules defined by system engineer. One of these configurations will then be simulated and implemented as a new production system.

V. Conclusion and Further Work

We introduced an approach to generate a set of configuration plans as the solutions for the implementation for the evolved production system according to the using of the I4.0 technologies on an ongoing production system. The approach was implemented within the scope of a Java environment and a clear implementation process.

We conducted two case studies to evaluate the approach focusing on the development risks evaluation and the economic evaluation. The case studies could show the applicability, efficiency, and suitability of the approach in example product systems. Moreover, development risks could be minimized by providing appropriate solutions for configuration plans. Furthermore cost estimations have turned out to be beneficial to optimize the overall costs by selecting an economic solution.

The underlying concept of managed evolution of production systems is currently being formalized including the formal descriptions and transformations. The results will be published in a future work.

Acknowledgement

This paper evolved of the research project "Synus" (Methods and tools for the synergetic conception and evaluation of Industry 4.0 solutions) which is funded by the European Regional Development Fund (EFRE — ZW 6-85012454) and managed by the Project Management Agency NBank.

References

[1] MCKINSEY DIGITAL, "Industry 4.0: How to navigate digitization of the manufacturing sector." [Online]. Available: https://www.mckinsey.de/files/mck_industry_40_report.pdf

[2] H. Giese, B. Rumpe, B. Schätz, and J. Sztipanovits, "Science and engineering of cyber-physical systems (dagstuhl seminar 11441)," Dagstuhl Reports, vol. 1, no. 11, 2012.

[3] H. Grönniger, J. O. Ringert, and B. Rumpe, "System Model-Based Definition of Modeling Language Semantics," Formal techniques for distributed systems, 2009, pp. 152–166.

[4] Institut der deutschen Wirtschaft Köln e.V. and VDI Verein Deutscher Ingenieure e.V., "Ingenieurmonitor 2019/I - Der regionale Arbeitsmarkt in den Ingenieurberufen." Institut der deutschen Wirtschaft Köln e.V., 2019. [Online]. Available: https://www.vdi.de/

[5] DZ BANK AG, "Mittelstand im Mittelpunkt - Ausgabe Frühjahr 2017." DZ BANK AG, Frankfurt am Main, 2017. [Online]. Available: https://www.dzbank.de/

[6] C. Stechert and H.-J. Franke, "Requirements Models for Modular Products," ICORD 09: Proc. of the 2nd International Conference on Research into Design, Bangalore, India, 2009.

[7] J. Schmitt, D. Inkermann, C. Stechert, A. Raatz, and T. Vietor, "Requirement Oriented Reconfiguration of Parallel Robotic Systems," Robotic Systems-Applications, Control and Programming, 2012.

[8] G. Simko, D. Lindecker, T. Levendovszky, S. Neema, and J. Sztipanovits, "Specification of Cyber-Physical Components with Formal Semantics – Integration and Composition," Specification of cyber-physical components with formal semantics - Integration and composition, vol. 8107 LNCS, 2013, pp. 471–487.

[9] T. Blochwitz et al., "Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models," in Proc. of the 9th International MODELICA Conference, no. 076, 2012, pp. 173–184.

[10] D. Wang, C. Knieke, and A. Rausch, "Data-driven Component Configuration in Production Systems," in Proc. of the ADAPTIVE 2019: The Eleventh International Conference on Adaptive and Self-Adaptive Systems and Applications. IARIA, 2019, pp. 44–47.

[11] [Online]. Available: https://isse.tu-clausthal.de/en/research/current-projects/synus/

[12] [Online]. Available: https://www.anylogic.com