Tackling Problems on Maintenance and Evolution in Industry 4.0 Scenarios Using a Distributed Architecture[1]

Sandro Koch[2] Tim Wunderlich[3] Jonas Hansert[4] Robert Heinrich[2] Thomas Schlegel[4]

Abstract: Growing complexity in Industry 4.0 environments goes hand in hand with an increasing number of vulnerabilities. Due to internal and external influences, these vulnerabilities can cause a negative effect on the production process as well as the finished product. These influences include e. g., defective machine tools, deviations in quality or configuration changes of an assembly line, which in turn require to adjust the model and the workflow of the respective facility to prevent risks to human health or minimise costs due to production stops. To evaluate the necessity of a novel approach to handle these vulnerabilities and influences along with their subsequent system adaptation, three emerging problems are identified and provide the basis for the discussion of a distributed architecture that aims to facilitate the evolution of context models, workflows and configurations while allowing for reasonable involvement of human operators.

Keywords: production system evolution; resilience; cyber-physical production systems; distributed systems

1 Introduction

Ubiquitous systems are becoming more and more prevalent in our everyday lives, especially in production. Integrated machines and self-organising production environments are examples of the increasing interconnection of humans, machines, and systems in the socio-technical production system of Industry 4.0. The more integrated and essential these systems become for the economy and society, the more important it is that the maintenance cycles and extensions for these systems can be carried out reliably and without sacrificing their availability. The increasing prevalence and availability of the Internet of Things (IoT) and associated IoT technologies create a much higher and more dynamic integration with the physical world for future software systems. Constant feedback of the software context with the relevant physical context at runtime thus enables sophisticated Cyber-physical systems (CPS). In this paper, we will address the problems that can arise in the development and maintenance of such an Industry 4.0 system. For this purpose, we present an illustrative example of a system in Sect. 2. In Sect. 3 we illustrate emerging problems on maintenance and evolution in Industry 4.0 scenarios. In Sect. 4 we present a proposal on how to solve the

[1] This work was partially funded by the German Federal Ministry of Education and Research under grant 01IS18067A/B/D (RESPOND), and the KASTEL institutional funding.
[2] Karlsruhe Institute of Technology, Am Fasanengarten 5, 76131 Karlsruhe, GER, sandro.koch|heinrich@kit.edu
[3] Fraunhofer IWU, Reichenhainer Straße 88, 09126 Chemnitz, GER, tim.wunderlich@iwu.fraunhofer.de
[4] Hochschule Karlsruhe, Moltkestr. 30, 76133 Karlsruhe, GER, jonas.hansert|thomas.schlegel@hs-karlsruhe.de

problems presented before. Finally, in Sect. 5 we outline future steps that are necessary to implement the solution ideas.

2 Motivating Example

Today's industrial plants often consist of several machines, which are often divided between different factory buildings. Nevertheless, in many cases, the complete production chain comes to a standstill if one machine fails. Often the machines in factories are not yet connected to the company's network. Edge devices provide the physical machines with access to the network and at the same time aggregate the often enormous amount of data that accumulates in the machines. In the following text, we call these edge devices nodes. In this way, we establish a close link between the physical plant and the software system, thus forming a Cyber-physical production system (CPPS) [Mo14].

The individual parts — such as factory buildings and machines — are represented as virtual nodes in a tree structure. The nodes can communicate with each other via events, and the parent nodes can combine the events of their children into more complex events. For example, events from different temperature sensors reporting too high temperatures could be merged by the parent node into an event reporting a fire in the machine. Also, there is a context component that uses all existing messages in the system to digitally map the real state of the plant. In addition to the errors detected directly by the nodes, other errors can also be detected in this way, such as a lost connection to a node. With a context meta-model [He18], further errors in the system can thus be modelled and detected. The context system is also used to find a self-healing strategy in the case of an error.

3 Emerging Problems on Maintenance and Evolution in Industry 4.0

As described in Sect. 2, several issues can occur when operating a CPPS. Those issues include e. g. machine outages, quality deviations, or human error, which potentially require the plant to undergo an adjustment regarding its component configuration or workflow management to minimise the incidence of these issues or to mitigate their effects. Due to the dependencies between the individual components of the overall system, changes in one component of the system could cause inestimable effects on other components [Kh11], similar to the ripple effect in software maintenance. With the aid of the Industry 4.0 scenario described in Sect. 2, we define three emerging problems, outlining possible maintenance and evolution issues in Industry 4.0 environments. We focus on adaptive and resilient Manufacturing Execution Systems (MES).

P1: Cope with Machine Failures and Resulting Downtime Today, typical manufacturing processes are highly automated and rely on machine tools, which are sometimes exceedingly specialised for one single manufacturing step, e. g. drilling or knurling. The quality of the end product depends on the consistent quality of the production process itself, which in turn depends on a complex interaction between production rate, system availability, defect rate, and other Key Performance Indicators (KPIs) [Co14]. Machine tools that fail to perform their tasks with the required quality — for example due to tool wear — result in defective products, which in turn could require rework or lead to rejection and therefore higher costs. In the worst case, a machine tool fails completely and cannot perform its task at all, causing a standstill of the whole production line. To minimise uncontrolled standstills as well as product rejects and the necessity for rework, defective components require a controlled downtime of the plant to provide a window of opportunity for maintenance, changeover, reconfiguration and replanning, which in turn increases costs and diminishes the efficiency of the manufacturing system [Li12].

P2: Cyber-Physical Deviation A complex manufacturing environment constitutes a network of CPS which can be described as a CPPS [Ne19]. Within such systems, sensors, actuators, and edge devices communicate with each other as well as with the business level and exchange data. Based on this data, models of the real world are generated on which the MES can make decisions about the manufacturing process. However, within a CPPS, deviations between the model and the reality can occur: For example, if the MES assumes that n workpieces are stocked, but in reality only $n - 1$ workpieces are available, the system again needs to provide a resilience strategy to compensate for this deviation. This includes updating the model and adjusting the manufacturing process, e. g. postponing another task that might need the same resources but is less prioritised, so it can allow for a reassignment of its resources. Furthermore, additional tasks in the overall process are required to restore a coherent CPS state, e. g. placing an order to stock up on the scarce resources.

P3: Human-in-the-Loop Despite the high degree of automation in manufacturing environments, manual labour can be still part of the overall processes (e. g. commissioning, packing, or maintenance), where errors are inevitable and cause an impact on the production and product quality [Bu05]. Since human errors can occur unanticipated, flexible resilience strategies are required and planned outages might not be sufficient for troubleshooting. To account for issues that occur in the context of manual tasks and attempt to rectify them on time, the resilient MES needs to fulfil two general requirements: On the one hand, it needs to keep track of specific actions performed by the respective person, and on the other hand, it needs to know how, when and to whom certain information needs to be delivered. Specific use cases include detecting an erroneous action performed by a worker: To fulfil an order, the worker needs to pick certain components. Utilising sensors and edge devices, the resilient MES would detect wrong pieces and notifies the worker about this issue. Depending on the equipment of the worker, the system decides how this notification will be delivered. [Fr16] describe how a linked-data platform can be used to manage and receive such traceability data via a standardised interface and subsequently provide a human worker with reasonable information in a context-aware manner. For example, a smart glove could be triggered to provide immediate haptic feedback and a smartwatch could display a message containing a description of the correct piece.

4 Modular Architecture for Industry 4.0 Scenarios

In this section, we propose an architecture as solution idea for the problems discussed in Sect. 3. The architecture covers not only an Automated Production System (aPS), but also the involved processes and systems which are directly or indirectly part of an Industry 4.0 system. We call the architecture presented here the RESPOND Architecture. RESPOND stands for Resilient socio-technical processes in the industrial Internet of Things.

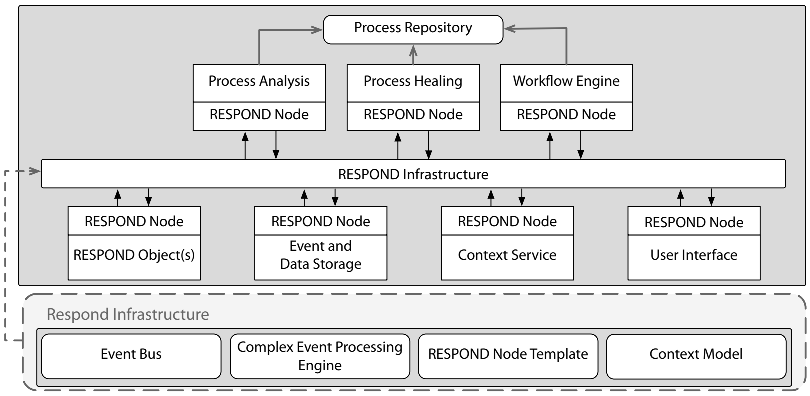

Fig. 1: RESPOND Architecture

Overview Fig. 1 shows the RESPOND architecture. An integral part of the architecture is the Industrial Internet of Things (IIoT) RESPOND node. A RESPOND node defines a component in the RESPOND system that exchanges data via the RESPOND infrastructure; this data does not necessarily have to be collected via a Programmable Logic Controller (PLC). This includes, for example, status and context information, but also raw data and aggregated or filtered values. Communication takes place via the RESPOND infrastructure, which integrates the RESPOND nodes and provides the platform functionality. Each RESPOND node implements a semantic profile, the RESPOND node template, which defines exchanged information and which context information is processed and generated. This ensures that the RESPOND node can be integrated into the Industry 4.0 system according to a specification in the sense of a Plug & Produce [DJ17]. Also, a RESPOND node must implement interfaces that enable the node to send and receive on the event bus, thus creating a Production Service Bus (PSB) [ST07]. This modular structure allows reducing possible downtime described in **P1** by focusing the maintenance on one RESPOND node instead of the whole RESPOND system. Since a RESPOND node abstractly describes a part of the Industry 4.0 system, any RESPOND node can be utilised as long as the specification is satisfied. In addition, the self-healing component should be able to adapt running processes to integrate already installed RESPOND nodes into the self-healing process, at least until the running process is finished, to prevent manual intervention. The effort to implement new features can also be reduced because of the modular structure of the architecture. In the example in Fig. 1, an Event Bus is provided by the RESPOND infrastructure for the transmission of information.

The RESPOND node is used to link different components, such as RESPOND objects, event storage, context service and a self-healing component with the RESPOND infrastructure. In the case of RESPOND objects, components of an aPS, such as motors, pumps, and sensors, are thus connected to the infrastructure. It should be emphasised that one node can be the interface for more than one object. Thus, existing elements of a plant can be combined and integrated into the RESPOND system. The RESPOND system is orchestrated by a Workflow Engine, which has access to a Process Repository, where the available processes are stored. If a node does not have the necessary computing capacity to aggregate events from different sources and filter them if necessary, the RESPOND Infrastructure provides the Complex Event Processing (CEP) Engine, which allows defining rules based on the events occurring in the RESPOND system. For example, the problems described in **P2** cannot be identified by only a plant, the data across different RESPOND Nodes need to be aggregated and interpreted. Extensive IIoT stream data is evaluated in addition to the event infrastructure in a data platform that provides the necessary time series evaluations for self-monitoring (Process Analysis) and self-healing (Process Healing). As described in **P3**, integrating the human into the equation generates the need for a resilient, error-prone system. Thus, our architecture provides a solution by integrating dedicated components for detecting unforeseen problems in the whole RESPOND system (Process Analysis) and inferring solutions (Self Healing) given the state and context of the Industry 4.0 system. The self-healing component should be designed in such a way that it can handle (planned) outages as well as unforeseen events on the basis of the process information and the abstract RESPOND node information. If, for example, a node of a production plant fails during production, production capacities can be reallocated by the self-healing component, so that the production process does not have to be stopped abruptly, but can continue with reduced throughput.

RESPOND Infrastructure The RESPOND infrastructure is the link between all nodes in an Industry 4.0 system. All connected nodes communicate with each other via an event bus. The Respond Template defines which interfaces a node must provide to be connected to the infrastructure. This ensures that despite the distributed approach, no incompatibilities arise between the nodes. If changes are necessary, due to the interfaces a node can be independently modified without changing the whole system. The context model in turn describes which elements can occur in an Industry 4.0 system [He18]. The concrete context model instance is derived from the nodes registered in the RESPOND system, the current system status is kept up-to-date via the context service and can be queried by other nodes in the RESPOND system if required. In the event of an error, nodes can be dynamically removed from the system; depending on the error-type, the CPPS does not have to go offline for this. A CEP-Engine is used to infer states across all nodes. Since the node of a plant or a plant itself cannot necessarily know about all events within an Industry 4.0 system, the CEP-Engine aggregates, filters and infers events and states that go beyond the scope of a node. This is necessary since, in the context of process analysis and process healing, correlations have to be considered which go beyond a fault at the node level.

5 Future Work

Our next steps are to realise the presented architecture using four use cases from the industry. This involves implementing the components described, such as the workflow management system, the RESPOND infrastructure, and the context service. In addition to information about the end-users, context information about processes, machines, resources and possible interaction technologies will be described semantically to create action and process-oriented solution concepts. Furthermore, user interfaces for resilient systems are being developed to ensure a consistent supply of information on the one hand and to support the various resilience mechanisms and check results for accuracy on the other.

References

[Bu05] Bubb, H.: Human reliability: A key to improved quality in manufacturing. Human Factors and Ergonomics in Manufacturing & Service Industries 15/4, pp. 353–368, 2005.

[Co14] Colledani, M.; Tolio, T.; Fischer, A.; Iung, B.; Lanza, G.; Schmitt, R.; Váncza, J.: Design and Management of Manufacturing Systems for Production Quality. CIRP Annals 63/2, pp. 773–796, 2014.

[DJ17] Dürkop, L.; Jasperneite, J.: Plug & Produce als Anwendungsfall von Industrie 4.0. In: Handbuch Industrie 4.0 Bd. 2. Pp. 59–71, 2017.

[Fr16] Friedemann, M.; Trapp, T. U.; Stoldt, J.; Langer, T.; Putz, M.: A framework for information-driven manufacturing. Procedia CIRP/, pp. 38–43, 2016.

[He18] Heinrich, R.; Koch, S.; Cha, S.; Busch, K.; Reussner, R.; Vogel-Heuser, B.: Architecture-based change impact analysis in cross-disciplinary automated production systems. Journal of Systems and Software 146/, pp. 167–185, 2018.

[Kh11] Khan, A.; Kastner, C.; Koppen, V.; Saake, G.: The pervasive nature of variability in SOC. In: 2011 Frontiers of Information Technology. IEEE, pp. 69–74, 2011.

[Li12] Liu, J.; Chang, Q.; Xiao, G.; Biller, S.: The costs of downtime incidents in serial multistage manufacturing systems. Journal of Manufacturing Science and Engineering 134/2, 2012.

[Mo14] Monostori, L.: Cyber-physical production systems: Roots, expectations and R&D challenges. Procedia CIRP 17/, pp. 9–13, 2014.

[Ne19] Neugebauer, R.; Ihlenfeldt, S.; Schließmann, U.; Hellmich, A.; Noack, M.: A new generation of production with cyber-physical systems–enabling the biological transformation in manufacturing. Journal of Machine Engineering/, 2019.

[ST07] Schlegel, T.; Thiel, S.: Semantic Models and Processes for Information Gathering and Interactive Workflow Execution in Decentralized Production Environments. In: Intelligent Production Machines and Systems – 3rd I* PROMS Virtual International Conference. 2007.