Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 41 (2016) 147 – 152

www.elsevier.com/locate/procedia

48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015

Context-sensitive adaptive production processes

C. Timurhan Sungur*[,b,c], Uwe Breitenbücher[b], Frank Leymann[b], Matthias Wieland[a]

[a]*Application of Parallel and Distributed Systems, University of Stuttgart, Universitätsstraße 38, 70569, Germany*
[b]*Insititute of Architecture of Application Systems, Stuttgart University, Universitätsstraße 38, 70569, Stuttgart, Germany*
[c]*Graduate School advanced Manufacturing Engineering, Stuttgart University, Nobelstraße 12, 70569, Stuttgart, Germany*
* Corresponding author. Tel.: +49 711 685 - 88 204; fax: +49 711 685 - 88 472. *E-mail address:* timurhan.sungur@iaas.uni-stuttgart.de

Abstract

To stay competitive, manufacturing companies need to adapt their processes in a regular basis to the most recent conditions in their corresponding domains. These adaptations are typically the result of turbulences, such as changes in human resources, new technological advancements, or economic crises. Therefore, to increase the efficiency of production processes, (i) automation, (ii) optimization, and (iii) dynamic adaptation became the most important requirements in this field. In this work, we propose a novel process modelling and execution approach for creating self-organizing processes: Production processes are extended by context-sensitive execution steps, for which sub-processes are selected, elected, optimized, and finally executed on runtime. During the election step, the most desired solution is chosen and optimized based on selection and optimization strategies of the respective processes. Moreover, we present a system architecture for modelling and executing these context-sensitive production processes.

Peer-review under responsibility of the scientific committee of 48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015

Keywords: Process; Automation; Optimisation; Adaptation

1. Introduction

Due to turbulences in their environments, manufacturing companies need to change their production systems regularly [1]. Following these changes, production processes need to be adapted. Moreover, customers demand more and more individualized products. This demand results in a vast number of process variants for each individualized product. Since it is not feasible to manually optimize each alternative control flow in the production process, an automated means for optimization is required to ensure an overall optimized production process [2]. Additionally, multiple execution steps can be executed simultaneously to increase productivity and precision of manufacturing processes [3]. The selection of each execution step may depend on different factors as new technological advancements provide more solution options to the same kind of problems. Especially, depending on the current situation of a production process, a selection from different solution options may be required to optimally execute the process. For instance, in production processes, humans are involved to conduct assembly tasks or to supervise the process [2]. Manually conducted assembly tasks may provide alternatives to the existing automation methods depending on the current demand, status of the machinery, and occupation of the machinery, i.e., the execution context. Thus, the selection of the right process alternative needs to consider the current situation in the respective factory.

Situations can be observed using cyber-physical systems of factories, which enable the application of well-adopted business process modeling and execution solutions in the context of manufacturing companies and tracking of activity flows in the real world [4]. Production processes can be modeled by activity sequences and structures using business process modeling languages, e.g., the Business Process Execution Language (BPEL) [5] or the Business Process Model and Notation (BPMN) [6]. After modeling, the process models are deployed on complaint workflow engines for an automated execution. However, in general, these process modeling and execution paradigms do not support context-sensitive adaptive modeling and execution of the business processes. By not considering the context-sensitive adaptation of production processes, the manufacturing companies lose

Peer-review under responsibility of the scientific committee of 48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015
doi:10.1016/j.procir.2015.12.076

their competitiveness by not reacting to changes in their environment on time.

In this work, we present a novel approach to support context-sensitive adaptive production processes. We extend production processes, which contain a sequence of predefined sets of sub-processes, with *Context-sensitive Execution Steps (CES)*. For each CES, context-relevant sub-processes are selected and desired processes ones are elected, optimized, and executed. The contributions of this paper can be listed as follows:

- Analysis of Production Processes (Sect. 4.)
- Context-sensitive Execution Steps (Sect. 5.)
- Context-sensitive Adaptive Production Processes (Sect. 6.)

In the following section, we present fundamentals and related work. Thereafter, we describe the motivating scenario used throughout this paper.

2. Fundamentals and Related Work

Business Process Management (BPM) life-cycles provide a systematic approach for modeling, execution, monitoring, and improvement of business processes. For example, the BPM Life-cycle of Weske [9] starts with the identification of business processes. During the modeling phase of business processes, modeling languages such as BPMN are used to capture recurring activity sequences and their structures. Similarly, these languages can be used to express production processes. For example, Zor et al. [7, 8] propose extensions to the BPMN-standard called BPMN4Manu to model manufacturing processes. Business process models are executed on complaint process execution engines [9].

Unlike many business processes, the execution of manufacturing processes typically depends on the information collected from the real world, i.e., the execution context. This *context* contains any information that may be useful to characterize situations of involved entities [10]. Internet of Things (IoT) technologies, such as RFID tags, wireless sensors, and wireless actuators, enable the systematic collection and manipulation of this context information [11]. By using these IoT technologies, it is possible to create *smart factories* that work autonomously based on the context information collected from production environments. Moreover, this context information can be managed using a context management middleware such as Nexus [12]. The collected context information drives the execution of business processes resulting in *smart workflows*. For example, we have proposed extensions of standard workflows with the context information provided by a smart factory [10, 4, 13]. For that purpose, we have developed three different constructs namely *context event*, *context query*, and *context decisions* [4]. Context events are triggered by context changes, context queries are used to query the status of a context, and context decisions are used to make decisions based on context information. Similarly, Wolf et al. [14] provide extensions to business processes to include context data created by IoT technologies such as sensors, mobile devices, etc. Different works have proposed to use declaratively defined activities, i.e., defining what needs to be done without defining how it is done, to enable context-driven execution of business processes. During the enactment, the actually needed steps to complete a declaratively-defined activity are selected based on the respective execution context [15, 16, 17]. Consequently, these processes become a context-aware process. Marella et al. [18] have proposed a similar approach using *business process planlets* for automated generation of alternative flows based on changes during execution. Hallerbach et al. [19] present an approach for modeling and managing process variants. In contrast to these activity-oriented approaches, Nurcan et al. [20] propose a strategy-driven modeling approach of processes. Processes defined based on the goals and refined to operational terms in the lower levels of the abstractions. Consequently, created models are easily changeable as they are decoupled from their operational terms. Such declarative approaches provide more flexibility and enable easier change of the business process models. For example, Van der Aalst et al. propose a declarative approach for modeling processes with constraints [21]. Processes modeled based on constraints allow various different executions. Similarly, we have presented a resource-centric declarative approach [22, 23]: Resources are associated with actors, which are responsible for the autonomous execution of an informally modeled process to achieve a specified intention. Barukh and Benatallah [24] propose a hybrid process management platform, on which structured to unstructured processes can be executed. The realization architecture in this work also enables execution of such hybrid processes using similar concepts.

As proposed by Zor et al. [8], we distinguish in this work between (i) manufacturing and (ii) production processes. Manufacturing processes only involve the processes to transform raw materials to products offered by the company whereas production processes contain both manufacturing and all relevant business processes to complete the production on time – for example, maintenance processes or raw material ordering processes. One can consider production processes defined at a macro-level whereas manufacturing processes are defined at a micro-level [25]. For instance, Azab and ElMaraghy [26] present a mathematical model for reconfigurable process planning for lower level technical processes, which can be incorporated in to manufacturing processes. To enable flexibility in manufacturing, the analysis of the changeability objects is needed as a first step [12].

3. Motivating Scenario

The motivating scenario is based on the case studies introduced by Erlach [27]. The scenario describes a production process that aims for the creation of glass panes. The manufactured glasses are robust security glasses used for the construction of buildings, doors, etc. Customers may order glasses with different specifications, e.g., different thickness, shape, drills, etc. The manufacturing process starts with a sub-process of cutting and shaping the raw glass. The glass is cut and panes are shaped based on customer request. After glasses have been cut, the production process continues with a sub-process of polishing glass panes. The polishing sub-process is followed by a drill and mill sub-process. Thereafter, the drilled and milled glass panes go through a washing and pretensioning sub-process for the generation of security glasses. The pretensioned glasses are moved to the quality control. Thereafter, the glass panes are delivered to the customers using different delivery processes. During the process execution, there are different points of adaptations. Each adaptation point requires collection of the context data and making decisions using the collected data. For example, the sub-process of cutting and shaping raw glass can be executed manually by human performers who cut the glass, or automatically by glass-cutting machines. The decision between automated and manual execution is done based on the current demand, the location of workers, and the availability of workers and machinery. Whenever there is a high demand and there are available workers in the workstation, the demand is suspended by cutting the glass manually. Similarly, the manual operation is preferred if glass-cutting machine is not functioning properly, which is the case once in a month. The number of resources employed for this sub-process depends on the demand and in case low demand, the cutting operation is completed using less number of machines. Moreover, a new method that introduces a semi-automated execution has been added to the production process as an alternative. Additionally, a glass cutting machine failure can results in a corresponding maintenance sub-process for the fastest and cheapest recovery.

When we analyze our motivating scenario, we can observe that, there are multiple *relevant process alternatives* that may be executed, e.g., an automated variant, a semi-automated variant, and a manual variant of glass cutting is possible. Consider the modeled example of the BPMN sub-process for cutting and shaping the raw glass shown in Fig. 1. Each context-dependent addition to this process requires a corresponding adaptation of the *Collect Context Data* activity and of the gateway, which can easily break the automated production process as models have to be (i) modified and (ii) tested, which is not easy if a sub-process is used in multiple other processes that may react differently to modifications due to different assumptions. Moreover, standard BPMN is not capable of handling context data, which results in the activity of *Collect Context Data* that pollutes the process model by data and context handling tasks that are not used to execute the actual production logic.

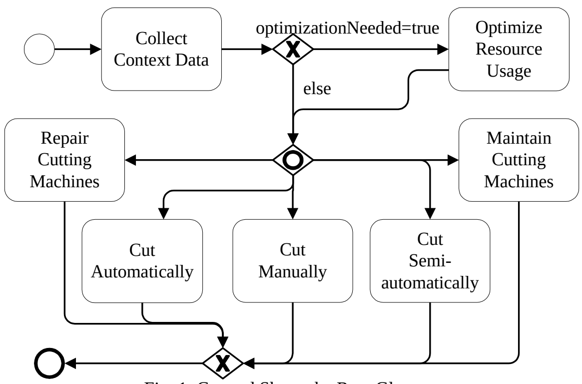

Fig. 1. Cut and Shape the Raw Glass

4. Analysis of Production Processes

In this section, we provide an analysis of typical production processes. We present a subset of drivers of manufacturing based on current trends and needs. Each driver is followed by a corresponding requirement which aims for improvement of production processes:

Driver 1 (D1): Spread of the Internet of Things. Technologies introduced by IoT trends enable building smart places, e.g., homes, buildings, cities, etc., by providing the runtime information of physical systems. The generated runtime data can be interpreted and necessary adjustments can be carried out based on the result of these interpretations. The application of these technologies to factories creates smart factories. Smart factories aim for adjusting their production processes based on produced information, e.g., by machines [28, 10, 29, 30, 12]. → R1: *Integration of the Internet of Things into Production Processes.* As a result of the D1, process definitions need to enable integrating the context information. Moreover, business process execution engines need to be made context-sensitive in a similar way [13, 4].

Driver 2 (D2): Context-driven Changeability of Manufacturing Systems. Smart objects of IoT technologies, e.g., wireless sensors and actuators, RFID chips, RFID sensors, etc., enable monitoring and analysis of the execution context created by technical processes [4]. Consequently, the processes can be aware of the execution context and adapt themselves based on the changes in the context. In this work, we call such an adaptation as *context-driven adjustability*. Moreover, in each such context definition, more than one process can be executed in parallel to increase productivity of the manufacturing processes [3]. → R2: *Context-driven Execution of Production Processes*. As a result, production processes need to support context-driven adjustability, which enables executing *more than one context-relevant* processes simultaneously, to increase the company's competitiveness.

Driver 3 (D3): Optimal Production Processes. Production processes aim for the individualization of their product portfolio, short development cycles, and high profit [28]. Consequently, an automated generation of optimal processes is desired [2]. The optimization of each different path in the execution is important for the overall optimization and changeability of the process [25]. → R3: *Optimizable Production Processes.* As a result of D3, production processes need to include relevant optimization strategies [31].

Driver 4 (D4): Goal-oriented Changeability of Manufacturing Systems. Technological advancements, existing demand, and various other turbulences can result in different alternative processes for the accomplishment of the very same goals, e.g., depending on the context of a request, a manual manufacturing can be favored although an automated alternative exists. Considering the trends towards automation, such a selection should be made automatically [28].

→ R4: *Goal-oriented Adaptation of Production Processes.* As a consequence, production processes need to realize such selection strategies so that they can be automatically shaped as the desired strategy of the respective production company.

5. Context-sensitive Execution Steps

To satisfy the requirements that have been introduced previously, we propose a new process modeling construct named *Context-sensitive Execution Step (CES)*. CES constructs are similar to the sub-process structures of BPMN, which are activated when the incoming flows are activated. Similarly, a CES construct defines input and output data. A meta-model of the CES concepts is shown in Fig. 3. To provide the *context-driven adjustability*, each CES contains a *Context Definition*. Each Context describes the properties of relevant things in the production environment, such as status of the glass cutting machines, location of the workers, and shape of the glasses in the motivating scenario. A Context Definition provides a mechanism to act adaptively based on the current situation in the production environment by describing each process for a specific Context Definition. For example, a CES for shaping the raw glass defines a manual sub-process if involved machines are not working properly and a mixture of an automated and manual sub-process for the context that all machines are available. Additionally, each CES specifies its *Main-Intention*, e.g., a goal of shaping the raw glass. Each Main-Intention can be refined by *Sub-Intentions*, which represent desired sub-goals of the execution step, e.g., high throughput, high-automation, high utilization, etc. For each Context Definition, there exists at least one *Realization Process* called *main realization processes* that aims for reaching the Main-Intention of the respective CES. For example, in the motivating scenario, there are three different main realization processes such as automated, manual, and semi-automated glass cutting that fulfill the Main Intention of the respective glass cutting and shaping sub-process. Moreover, there may exist Realization Processes with other Main-Intentions under a Context Definition, i.e., *complementary realization processes*, such as the maintenance sub-process from the motivating scenario. The Main Intentions of the Realizations Processes are refined similarly with Sub-Intentions. Each Intention can contain a *Selection Strategy* for choosing between multiple processes with the same goals, e.g., a selection strategy for glass cutting and shaping intention chooses between the automated and

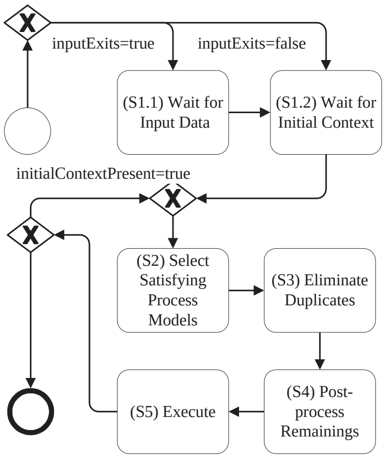

Fig. 2. Operational Semantics of a CES

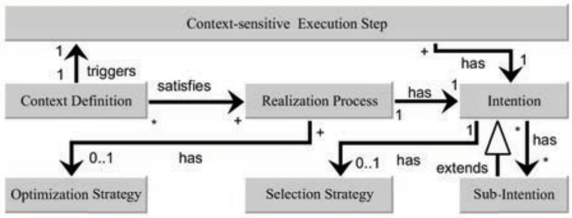

Fig. 3. Context-sensitive Execution Step Meta-Model

manual cutting using the execution context data. In case no Selection Strategy has been provided, a random selection is done. As all random selection candidates aim for achieving the same goal, it is guaranteed that the selection also works towards the accomplishment of the desired Main-Intention. These strategies are created based on the respective priorities of the organizations. For example, an organization may be in the favor of a process with a more energy consumption and with a more efficiency. Each realization process can contain an *Optimization Strategy* to provide means of an *automated* process-specific optimization depending on the execution context. For example, in the motivating scenario, we have the activity of optimizing resources before executing the glass cutting operation. Each Realization Process can be defined as a *Context-sensitive Production Process*, which enables different levels of granularity during modeling.

In a production process, after reaching each CES construct, the execution follows the steps shown in Fig. 2. A CES can be defined with input and output data. If that is the case, the input data is awaited at first (S1.1), which arrives typically at the activation of a CES construct. The relevant realization process definitions may use the provided input data or use the available context data. However, they must not specify any data that does not exist in the execution environment, which would result in an invalid model that is rejected by the runtime.

After receiving the input data, the Context Definition of the CES needs to be present in the execution environment to continue with the next step (S1.2). If no Context Definition has been specified, the process behaves as if the Context Definition has been directly evaluated to true, i.e., the default value of a context definition is true and the execution proceeds with the *Select Satisfying Process Models* step.

During this selection step (S2), the Context Definitions of all available Realization Processes are evaluated. The execution of a CES construct stays active as long as the context defined by this CES is present in the execution environment. The Realization Processes, whose Context Definitions are present in the environment, are selected for further execution steps. In this set, there must be at least one main realization process; otherwise a run-time error is thrown.

After selecting all Realization Processes, the CES execution continues with the *Elimination of the Duplicates* step (S3). Each selected Realization Process specifies a Main-Intention, which may be detailed with Sub-Intentions. Whenever there are more than one Realization Process for reaching a specific Main-Intention, one of them is elected using their respective Selection Strategies. When no Selection Strategy is defined, a random selection is done. As a result,

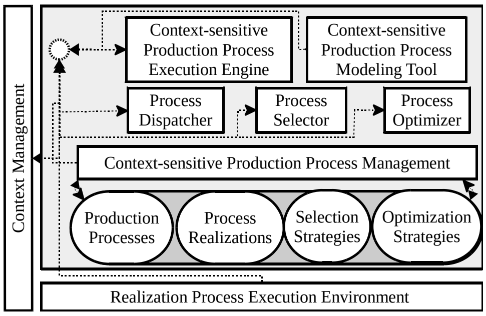

Fig. 4. Realization Architecture

from various options the desired one for the organization is selected based on the specified Selection Strategy.

Before executing each Realization Process, their respective Optimization Strategies are executed (S4). The resulting processes are considered to be the best possible execution and each realization process is executed. The Realization Process that has the same Main-Intention as the CES construct must be operationally compatible, i.e., they must produce the same output data as the CES construct. After each completion of a main realization process, the enactment of consecutive activities of the respective CES begins. If any complementary realization process terminates with an error, the execution is assumed to be successful as the main realization process has succeeded. However, in such a case the respective CES completes with a warning.

6. Context-sensitive Adaptive Production Processes

The presented CES concept has been introduced as a separate entity although they are part of a production process. To integrate CES constructs, we extend *Standard Production Processes* to *Context-sensitive Adaptive Production Processes* by inserting CESs. As a result, the meta-model has a backward compatibility with the state of the art concepts. As mentioned in the fundamentals section, changeable manufacturing is enabled by the identification of changeability objects in manufacturing. For that purpose, we append the process identification activity of the BPM life-cycle with an additional step of analyzing changeable process parts. These changeable parts are defined as CES constructs in production processes. Each CES is defined using a context definition, i.e., the initial context of the corresponding CES. A main intention is identified for each CES. Moreover, the input and output data of the CES construct is specified. Thereafter, Realization Processes are added to a central repository so that the execution of different CES constructs can check available processes. There must be a Realization process with the same main intention satisfying each CES. The realization processes with the same main intention, as the parent CES construct should provide compatible functional interfaces. Finally, the modeled context-sensitive adaptive production process is deployed.

We are developing a system to realize the introduced concepts. The architecture of our realization is depicted in Fig. 4. Processes with the CES constructs are modeled using a custom *Context-sensitive Production Process Modeling Tool* component that enables the addition and editing of CES

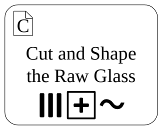

Fig. 5. CES Construct

constructs. For instance, the respective CES construct using BPMN notation is represented in Fig. 5.

Each sub-process follows the steps in Section 5 during the execution. The modeling tool should exploit the available context information provided by the respective *Context Management* component. Consequently, during modeling of CES constructs, only valid context data is allowed, which would decrease the chance of runtime errors. After modeling of a context-sensitive adaptive process is completed, it is deployed on *Context-sensitive Production Process Execution Engine,* which supports the modeled CES constructs. Whenever the process execution reaches to a CES construct, it compares desired initial context of the CES construct with the present context information gathered from *Context Management* component. When the present context matches the context expression in the CES construct, the *Context-sensitive Production Process Execution Engine* component, calls *Process Selector* with the goal definitions of the CES. The *Process Selector* executes the steps of selection and election defined in Sec. 5.2. It uses *Selection Strategies* for electing and selecting the matching realization processes with the same main intentions. Thereafter, the Process Optimizer executes the Optimization Strategies of the selected processes in case they have such a strategy. All realization processes after the step of optimization passed to the Process Dispatcher component for the execution. This component interacts with Realization Process Execution Environment to deploy and execute the realization process. This environment can be another process execution engine or a manufacturing machine. The *Process Dispatcher* is responsible for the life-cycle operations of each enacted realization process. The components wait all the dispatched processes to be executed, and returns control to the *Context-sensitive Production Process Execution* engine. The *Context Driven Context Management* component manages all process artifacts.

7. Evaluation of the Approach and Related Work

The integration of the IoT concepts (R1) are considered by the proposed approach, as the activation and execution of the CES construct depend on the execution context. The process constructs from our previous work also enable inclusion of execution context during modeling of business processes [4]. Similarly, Wolf et al. [14] provide necessary constructs to use context in business processes. Similarly, the declarative activity approaches [15, 16, 17] include context information similarly. On the other hand, the rest of the analyzed related work does not address R1.

The Context Definition of CESs enable the execution of processes driven by the present context data. Thus, the proposed approach satisfies R2. Similarly, the context-events proposed in our previous work [4] can be used to take decisions based on the available context information.

Declarative activity approaches [15, 16, 17] enable context-driven adaptation of the process. However, the execution of multiple relevant process under a certain context definition is not addressed. Consequently, they meet R2 only partially. The other described related work does not address R2 at all.

As the conceptual model in Fig. 3 shows, we enable inclusion of *Optimization Strategies* for each process that may be executed later on. As each variant process is optimized using its predetermined optimization strategy, the approach satisfies requirement R3. Such an optimization is not considered by the other described related work. Furthermore, by providing the *Selection Strategies* for each *Intention*, we enable goal-oriented adaptation for each relevant process. Thus, our approach meets R4. Declarative activity approaches [15, 16, 17] enable selection of the processes satisfying the goal of an declarative activity. As the parallel execution of multiple relevant processes is not considered, such a goal-oriented adaptation is not addressed and they fail to meet R4.

8. Conclusion and Outlook

In this work, we have presented a new workflow modeling construct the Context-sensitive Execution Step (CES). Based on that we enable an extended modeling approach that allows to model context-aware production processes Furthermore, we presented a system architecture providing an optimized execution of these context-aware production processes that adapt to their execution environments based on the preferences of the respective companies. In the future work, we will create a case study based on these concepts in which we integrate manual and automated processes.

Acknowledgements

This work is supported by Graduate School of Excellence advanced Manufacturing Engineering (GSaME)[1] and partially received funding from the German government funded DFG project SitOPT (Research Grant 610872).

References

[1] Westkämper E. Factory Transformability: Adapting the Structures of Manufacturing. In: Reconfigurable Manufacturing Systems and Transformable Factories. Springer; 2006.
[2] Schlick CM, Faber M, Kuz S, Bützler J. A Symbolic Approach to Self-optimisation in Production System Analysis and Control. In: Advances in Production Technology. Springer International Publishing; 2015.
[3] Lauwers B, Klocke F, Klink A, Tekkaya E, Neugebauer R, McIntosh D. Productivity Improvement Through the Application of Hybrid Processes. In: Advances in Production Technology. Springer International Publishing; 2015.
[4] Wieland M, Kopp O, Nicklas D, Leymann F. Towards Context-Aware Workflows. In: CAiSE'07 Proceedings of the Workshops and Doctoral Consortium Vol.2; 2007.
[5] Web Services Business Process Execution Language Version 2.0; 2007. Available from: http://docs.oasis-open.org/wsbpel/2.0/OS/-wsbpel-v2.0-OS.pdf.
[6] Object Management Group (OMG). Business Process Model and Notation Version (BPMN) Version 2.0 Specification; 2011. Available from: http://www.omg.org/spec/BPMN/2.0/PDF/.
[7] Zor S, Görlach K, Leymann F. Using BPMN for Modeling Manufacturing Processes. In: Proceedings of 43th CIRP International Conference on Manufacturing Systems; 2010.
[8] Zor S, Schumm D, Leymann F. A Proposal of BPMN Extensions for the Manufacturing Domain. In: Proceedings of 44th CIRP International Conference on Manufacturing Systems; 2011.
[9] Leymann F, Roller D. Production Work Flow: Concepts and Techniques. Prentice Hall PTR; 2000.
[10] Wieland M, Leymann F, Schäfer M, Lucke D, Constantinescu C, Westkämper E. Using Context-aware Workflows for Failure Management in a Smart Factory. In: UBICOMM 2010; 2010.
[11] Atzori L, Iera A, Morabito G. The Internet of Things: A survey. Computer Networks. 2010;54:2787–2805.
[12] Lucke D, Constantinescu C, Westkämper E. Smart factory - a step towards the next generation of manufacturing. In: Proceedings of the 41st CIRP Conference on Manufacturing Systems; 2008.
[13] Wieland M, Kaczmarczyk P, Nicklas D. Context Integration for Smart Workflows. In: Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications; 2008.
[14] Wolf H, Herrmann K, Rothermel K. Modeling Dynamic Context Awareness for Situated Workflows. In: OTM 2009 Workshops. Springer Berlin Heidelberg; 2009.
[15] Andrikopoulos V, Bucchiarone A, Gómez Sáez S, Karastoyanova D, Mezzina C. Towards Modeling and Execution of Collective Adaptive Systems. In: ICSOC 2013 Workshops. Springer International Publishing; 2014.
[16] Bucchiarone A, Marconi A, Pistore M, Raik H. Dynamic Adaptation of Fragment-Based and Context-Aware Business Processes. In: ICWS 2012; 2012.
[17] Adams M, ter Hofstede A, Russell N, van der Aalst WMP. Dynamic and Context-Aware Process Adaptation. In: Handbook of Research on Complex Dynamic Process Management: Techniques for Adaptability in Turbulent Environments. Business Science Reference; 2009.
[18] Marrella A, Russo A, Mecella M. Planlets: Automatically Recovering Dynamic Processes in YAWL. In: On the Move to Meaningful Internet Systems: OTM 2012. Springer Berlin Heidelberg; 2012.
[19] Hallerbach A, Bauer T, Reichert M. Capturing variability in business process models: the Provop approach. Journal of Software Maintenance and Evolution: Research and Practice. 2010.
[20] Nurcan S, Etien A, Kaabi R, Zoukar I, Rolland C. A Strategy driven business process modelling approach. BPMJ. 2005;11:628–649.
[21] van der Aalst WMP, Pesic M, Schonenberg H. Declarative workflows: Balancing between flexibility and support. CSDR. 2009;23:99–113.
[22] Sungur CT, Kopp O, Leymann F. Supporting Informal Processes. In: ZEUS 2014; 2014.
[23] Sungur CT, Binz T, Breitenbücher U, Leymann F. Informal Process Essentials. In: EDOC 2014; 2014.
[24] Barukh M, Benatallah B. ProcessBase: A Hybrid Process Management Platform. In: Service-Oriented Computing. Springer Berlin Heidelberg; 2014.
[25] Wiendahl HP, ElMaraghy HA, Nyhuis P, Zäh MF, Wiendahl HH, Duffie N, et al. Changeable Manufacturing - Classification, Design and Operation. CIRP Annals. 2007;56:783–809.
[26] Azab A, ElMaraghy H. Mathematical modeling for reconfigurable process planning. CIRP Annals-Manufacturing Technology. 2007.
[27] Erlach K. Value stream design: the way towards a lean factory. Springer; 2013.
[28] Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M. Industry 4.0. BISE. 2014;6(4):239–242.
[29] Schuh G, Reuter C, Hauptvogel A, Dölle C. Hypotheses for a Theory of Production in the Context of Industrie 4.0. In: Advances in Production Technology. Springer International Publishing; 2015.
[30] Drath R, Horch A. Industrie 4.0: Hit or Hype? Industrial Electronics Magazine, IEEE. 2014 June;8(2):56–58.
[31] Klocke F, Abel D, Hopmann C, Auerbach T, Keitzel G, Reiter M, et al. Approaches of Self-optimising Systems in Manufacturing. In: Advances in Production Technology. Springer International Publishing; 2015.

[1] http://www.gsame.uni-stuttgart.de/