This is a pre-print of an article published in Research in Engineering Design. The final authenticated version is available online at: https://doi.org/10.1007/s00163-018-0296-5

A Systematic Approach for Supporting the Adaptation Process of Discrete Manufacturing Machines

Philipp Marks[+1], Xuan Luu Hoang*, Michael Weyrich[+], Alexander Fay*

[+] University of Stuttgart
Institute of Industrial Automation and Software Engineering
Pfaffenwaldring 47, 70550 Stuttgart, Germany

[*] Helmut Schmidt University/University of the Federal Armed Forces Hamburg
Institute of Automation Technology
Holstenhofweg 85, 22043 Hamburg, Germany

Abstract

Automated manufacturing machines in the discrete manufacturing domain are frequently facing changes in environmental conditions such as volatile customer demands or changes in product variants. Due to this, machines need to become more flexible to cope with these changing conditions. Therefore, manufacturing machines have to undergo adaptation processes during their operational phase. The adaptation processes might include mechanical, electrical and software changes. In industrial practice, these adaptation processes are individually performed by experts without methodological support which is time-consuming and highly error-prone. This article proposes a structured approach for supporting the different phases of the adaptation process. The producibility check of a production request based on a suitable skill model of the system is addressed as well as the automatic generation of adaptation options. Furthermore, the article provides concepts for analyzing the impact, effort and benefit of the generated adaptation options. Additionally, a multi agent architecture is presented for the implementation of the proposed adaptation approaches. The entire assistance concept was applied to a lab-size production machine to validate the applicability of the approach.

Keywords

Adaptation; Discrete Manufacturing; Manufacturing Machines; Machine Lifecycle; Assistance Concept; Manufacturing Flexibility

Acknowledgement

The German Research Foundation (DFG) funds this research within the research project “FlexA” (FA 853/9-1 and WE 5312/5-1). We express our sincere thanks to the DFG for the generous support of the work described in this article.

---

[1] Corresponding Author:

philipp.marks@ias.uni-stuttgart.de
Tel.: +49 711 685 67293
Fax: +49 711 685 67302

1 Introduction

Automated manufacturing machines in the discrete manufacturing domain are usually constructed for an operation period of several years or decades, due to their high investment costs (Birkhofer et al. 2012). Nowadays, during this period the manufacturing requirements and environmental conditions are frequently changing due to global trends, such as volatile customer demands, shorter product lifecycles and highly variable product portfolios (ElMaraghy and AlGeddawy 2012; Vogel-Heuser et al. 2015; Ladiges et al. 2016). For this reason, manufacturing machines need to possess a certain degree of flexibility to handle these changes and maintain fully operative throughout their entire operational phase. The life cycle of manufacturing machines can divided into different phases, namely conception, design & development, manufacturing & set-up, operation & maintenance and disposal phase (Enparantza et al. 2006).

In practice, the flexibility of manufacturing machines is designed to cope with a predefined set of requirements and environmental conditions. This set has been usually determined at early engineering stages and can also include future predicted changes of requirements and conditions. However, once changes occur which have not been anticipated in the set, the initial designed flexibility of a machine is insufficient to handle these changes. Thus, the machine cannot continue to operate and becomes obsolete. Because of the long operational phase of automated manufacturing machines, a high percentage of operating machines in companies are legacy machines. These machines mostly have been designed for a more static manufacturing environment and, accordingly, provide a lower degree of flexibility. Consequently, legacy machines are highly sensitive towards changes and can rapidly become obsolete. This reduces the actual operational phase and profit of the machine. To avoid this problem and to keep the machines profitable, legacy machines have to be adapted in accordance with the changed requirements and conditions (Rogalski 2011).

In contrast to legacy machines, newly designed machines normally are constructed with a larger flexibility or adaptability, as they are supposed to operate in a dynamical environment. Accordingly, this type of machines can better cope with changes. Based on the standard VDI 5201, in this contribution the flexibility of a manufacturing machine is regarded as the range of a certain indicator, e.g. output rate which the machine can perform without changing its current structure. In contrast, the adaptability of a machine is described by the ratio between the time of a structural change compared to the increase of the flexibility corridor bandwidth. The relationship between flexibility and adaptability is illustrated in Figure 1. Paradigms like Flexible Manufacturing Systems (ElMaraghy, 2005) or Holonic Manufacturing Systems (Brennan and Norrie, 2003) aim at integrating flexibility or adaptability in the manufacturing machine. Still, the degree of flexibility and adaptability that such manufacturing systems can offer is determined in the design phase of the manufacturing system. Design for adaptability (Kasarda et al., 2007) is a methodology that aims at achieving advanced sustainable designs that take changing requirements into account. Engel and Browning (2008) have extended this methodology and combine it with the concept of real options from economics to derive and evaluate architecture options. In their work, modularity is seen as a key factor for adaptability (Engel and Reich, 2015). However, they also state that a higher modularity is increasing the inter-module interface costs. In order to determine an optimal level of adaptability for a product's architecture, the authors introduce an architecture adaptability value which describes the balance between the modularity level and interface costs. The concepts have been approved in several industrial case studies which showed the applicability of the architecture adaptability value concept (Engel et al. 2016). Another approach which aims at generating flexibility in the design of engineering systems has been presented by Hu and Cardin (2015). In their approach the authors try to identify critical system elements that are related to the flexibility by identifying dependencies between elements and sources of uncertainties, which could cause changes. Subsequently, a risk measurement is determined using a Bayesian network model and normalized costs of probable changes. Based on the calculated risk, recommendations for the redesign of the system are proposed to increase the flexibility. However, the outlined methodologies are not describing systematizations for the adaptation of machines but rather design concepts for the creation of machines with a higher adaptability or flexibility. Furthermore, machines that were designed using the previously described methodologies also have to be adapted to avoid degeneration, if requirement changes occur that have not been predicted in the design phase. Hence, adaptations ought to be regarded as a natural part of the machine's life-cycle as opposed to as an extraordinary action (Tompkins 2010).

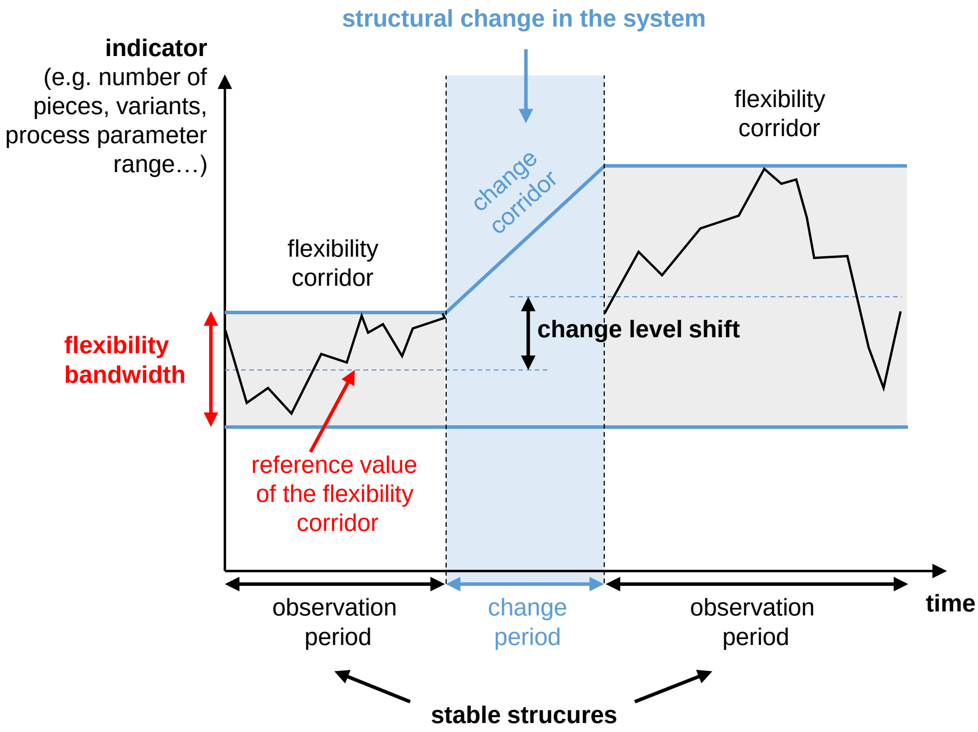


Fig. 1 Illustration of flexibility over time based on VDI 5201

An adaptation comprises changes in the physical structure of a machine and/or its automation software and involves different engineering disciplines (Ladiges et al. 2013a). Accordingly, several interdependencies must be considered and analyzed while conducting adaptation actions in order to obtain the desired changes and avoid unintentional changes of the system behavior (Ladiges et al. 2013b). Thus, the adaption process is a complex and tedious procedure. However, adaptation processes are currently mostly performed manually and individually for each case, as no assistance system or standardized process exists. Thus, the conducted adaptations are time-consuming, error-prone and depend exceedingly on the knowledge of the operating staff. A systematized and standardized approach for adaptations in combination with an intelligent assistance system could guide engineers through this complex process and would result in a more efficient and predictable adaptation process.

Single aspects of the adaptation process have been investigated in the literature, e.g. the automated documentation of changes between feature models (Bürdek et al. 2016) and the application of delta modelling to describe software changes (Seidl et al. 2017). Verification support for the software evolution is presented by Ulewicz et al. (2016) and Wenzel et al. (2014). However, the investigated approaches have been developed for the adaptation of automated manufacturing machines in the design phase and do not focus on adaptations in the operational phase. A work which addresses the adaptation support in the operational phase of automated manufacturing machines has been presented by Haubeck et al. (2013). However, this work was focused on the adaptation of manufacturing automation software and not on the adaptation of hardware. A detailed overview of investigated approaches and their applicability in the different phases of the adaptation process and the life phase of the manufacturing machine will be provided in Section 3.

Accordingly, the aim of this contribution is to propose a systematized approach for the adaptation of existing automated manufacturing machines in their operation phase since a lack of methodologies to support the adaptation process in this phase has been identified. Since adaptation actions are part of the lifecycle of manufacturing machines, the available approaches for integrating flexibility and adaptability in the design phase of a manufacturing machine are not sufficient anymore. The approach described in the following addresses the entire adaptation process, presenting different methods for each process phase. Furthermore, this contribution introduces an agent-based assistance system that utilizes the developed methods to support the engineer during the adaptation process.

The rest of the article is organized as follows: Section 2 introduces and discusses the different phases of the adaption process and their challenges. Subsequently, a review of related works and existing approaches for the identified challenges is presented in Section 3. Section 4 presents the systemized approach for the adaption process, including the different methods for each phase. This is followed by the implementation concept of the agent-based assistance system which is introduced in Section 5. Section 6 illustrates the validation of the proposed concepts by application on a lab-size production system. Finally, in Section 7 the article closes with conclusions and an outlook on future work.

2 Adaptation Process of automated Manufacturing Machines

The adaptation of manufacturing machines is a knowledge-driven task that is mostly performed by experts (Marks et al. 2017) of different domains. Since automated manufacturing machines are mechatronic systems, the disciplines involved normally include mechanical, electrical and software engineering. Interdependencies between the domains have to be considered in order to generate and evaluate adaptation options. The knowledge of the experts as well as the time available for the planning of possible adaptation solutions limits the quality of the solution (Beyer et al. 2016). Since most manufacturing machines are one-of-a-kind-systems, the adaptation process is performed individually for each machine which is time-consuming and error-prone (Hoang et al. 2016).

Although a common understanding of the term exists, there is no consistent description in the literature of the adaptation process of manufacturing machines (Koch et al. 2016). In this contribution, the adaptation process will be divided into four major phases that are depicted in Fig. 2. The phases will be used in the following sections to describe the assistance concept for each specific phase and are also valid for the manual execution of the adaptation process by experts.

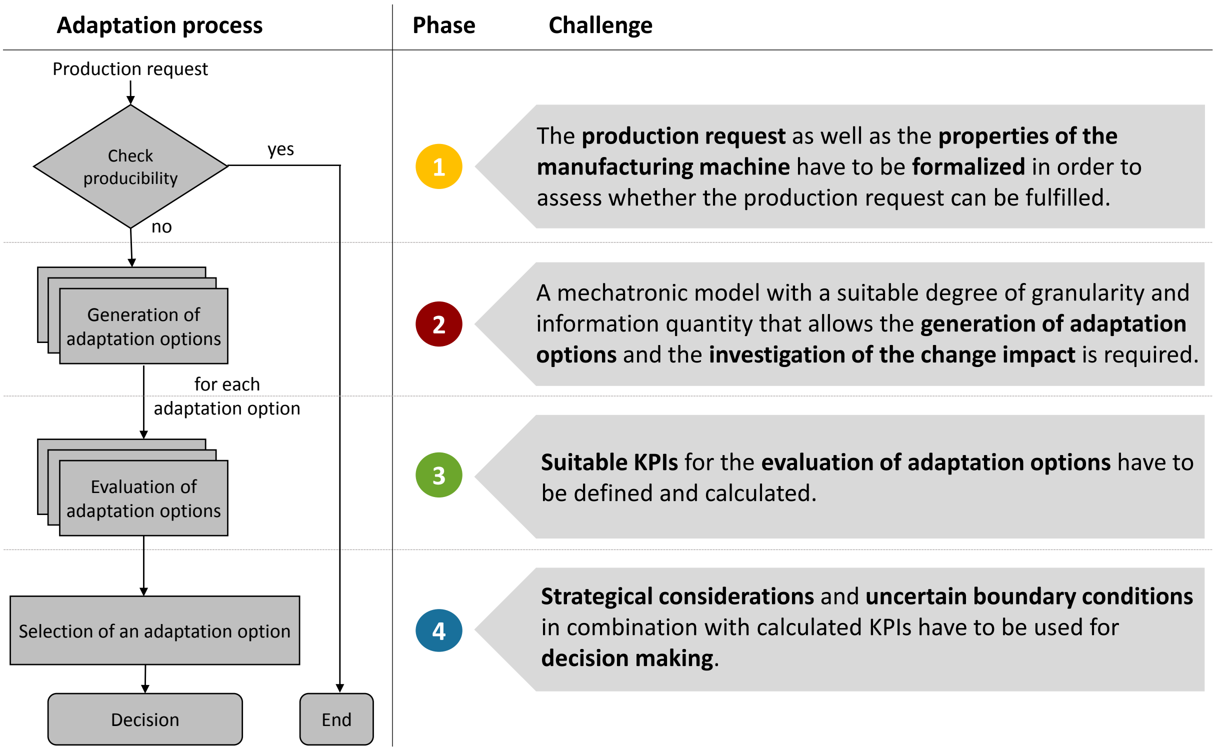


Fig. 2 General sequence and challenges of the adaptation process

In the first phase of the process, it has to be determined whether a given production request can be fulfilled by the current manufacturing system. The production request includes properties and requirements of the product that shall be manufactured as well as requirements for the production process itself (e.g. tolerances, speed). The production request is normally given in an informal way but needs to be transferred into a formal description if a computer-based assistance concept shall be applied. If the performed checks of the first phase conclude that the production request can be fulfilled, i.e. that the product can be manufactured, no adaptation action is required.

Otherwise, feasible adaptation options have to be generated in the second phase of the process. The generated adaptation options have to consider interdependencies between system elements (e.g. components, modules) and the different engineering domains. Due to the mechatronic nature of automated manufacturing machines, a change in one discipline usually implies changes in other disciplines. For the automatic generation of adaptation options, a mechatronic model of the system is required. Challenges for the model are the determination of the right level of granularity (Maga et al. 2011) and the reduction of the effort needed to model the production system. The generation of adaptation options in current adaptation projects is limited to time and knowledge of the engineer(s) performing this task. A computer-supported approach could allow the generation of a greater number of adaptation options ensuring a certain level of quality of the results.

In the third phase of the adaptation process, the previously generated adaptation options have to be analyzed and evaluated. The evaluation considers different criteria such as time or cost needed for the adaptation as well as benefits of the proposed solution, e.g. in terms of increased flexibility. The result of this cost-benefit analysis can be used as a basis for the selection of one of the proposed adaptation options. The main challenge of the third phase is the selection of suitable key performance indicators (KPIs) that support the subsequent decision-making process.

This decision-making process takes place in the fourth and last phase of the adaptation process. Based on the previously determined KPIs, one of the proposed adaptation options has to be selected as the solution that shall be implemented. The main challenge in this phase is that strategical considerations and uncertain boundary conditions have to be taken into account. After the selection of an adaptation option, the necessary changes are performed on the automated manufacturing machine, usually starting by the adaptation of the hardware, followed by the adaptation of the software. The decision-making process is not regarded further in this article since it requires more than only the technical view on the manufacturing machine. Suitable approaches for this phase can be found in the field of multi criteria decision making and in the field of multi agent systems which are able to deal with uncertain environments. In the authors' opinion, this phase always requires a human-in-the-loop and can only slightly be supported by an assistance concept.

As one can see, different challenges occur when the adaptation process shall be supported by an assistance concept. These challenges are summarized on the right side of Fig. 1. This contribution addresses the challenges of the first three phases and gives an overview of existing approaches for these challenges in Section 3. Based on the literature review and the identified shortcomings, concepts for each phase of the adaptation process have been developed (Section 4) and integrated into an overall assistance concept (Section 5) for the adaptation of automated manufacturing machines.

A first step towards a systematization of the adaptation process of manufacturing machines, which also applies to handling machines, was presented in Hoang et al. (2016). The approach is based on two aspects: 1) manufacturing capability characteristics of machines and 2) adaptation categories. The four characteristics that define the manufacturing capability namely are: (C1) set of different manufacturing operations, (C2) parameter range of manufacturing operations, (C3) set of feasible sequences of manufacturing operations and (C4) range of output quantity. To extend the functionality of the machine, actions of three adaptation categories or a combination thereof can be performed:

- Category A: Integration of manufacturing operations
- Category B: Adaptation of operation parameter ranges
- Category C: Adaptation of material flow

This contribution will mainly focus on characteristics C2 and partly C4 and will propose adaptation options from category B. Characteristics C1 and C3 are only treated superficially since this would extend the scope of this article. Thus, this contribution puts its focus on the adaptation and flexibilization of process parameter ranges of manufacturing machines. Note here that in the following, manufacturing operations which a machine can accomplish are referred to as skills of this resource.

3 State of the Art

This section presents a review of related work that addresses the previously described challenges of the first three phases of the adaptation process of automated manufacturing machines. A summary of investigated approaches and their applicability in the first three phases of the adaptation process is given in Table 1.

Table 1: Overview of investigated approaches and applicability in the first three phases of the adaptation process (white cell background: not applicable, light grey: partly applicable, dark grey: fully applicable)

Approach		Adaptation phase

	Considered life phase	Check producibility	Generation of adaption options	Evaluation of adaptation options
Engel and Browning (2008), Engel and Reich (2015), Engel et al. (2016)	Design			
Allen et al. (2016)	Design			
Järvenpää et al. (2016)	Operation			
Pfrommer et al. (2015)	Operation			
Cavin and Lohse (2014)	Operation			
Clarkson et al. (2004)	Operation			
Karl and Reinhart (2015)	Operation			
Fei et al. (2011)	Operation			
Ollinger and Stahovich (2004)	Operation			
Yang and Duan (2012)	Operation			
Ahmad et al. (2013)	Operation			
Zäh et al. (2011)	Operation			
Zhang et al. (2017)	Design, Operation			
Das et al. (2000)	Disposal			
Lucas and Tilbury (2005)	Development			
Gharieb (2006)	Development			
Brill and Mandelbaum (1989), Buzacott and Mandelbaum (2008)	Operation			
Kochikar and Narendran (1992), Das (1996), Wahab et al. (2008)	Operation			
Proposed Approach	Operation			

Legend:
Not applicable
Partly applicable
Fully applicable

3.1 Producibility check approaches

In the literature, several approaches have been proposed for the modeling of resource skills (also referred to as resource capabilities) in the field of production. These skill descriptions are utilized in order to check if a certain production request can be manufactured on a machine. Järvenpää et al. (2016), for instance, introduced a formal capability model that is used for the comparison of product requirements with resource capabilities, in order to orchestrate them. In their approach, capabilities are specified by a name and parameters. Complex capabilities are modeled by the combination of atomic capabilities. For the skill description, the authors employ an ontological description that is based on the CoreOntology presented in Jarvenpää (2012). Another skill description concept for the comparison of product requirements and resource capabilities has been proposed by Pfrommer et al. (2015). In the concept, skills are described as the ability of a resource to perform a technical process. Accordingly, the skill of a resource is described by process parameters and their values. In order to compare product requirements with resource skills, the concept requires a bill of processes of a product. Based on this bill of processes, the required processes are matched against the skill description of the resources. An approach that aims in the same direction has been presented by Cavin and Lohse (2014). Here, resource skills are described by their executable processes on different granularity levels which allows the matching of product requirements and resource skills on different levels. The approach is also based on the notion that product requirements are described by a bill of processes.

The outlined approaches have in common that their description concepts for resource skills are all based on the technical process. Furthermore, the concepts heavily depend on the bill of processes in order to match product requirements and skills of resources. Due to these characteristics, the outlined concepts have several shortcomings. For instance, the modeling complexity of resource skills in these concepts can be rather high, as the concepts have to model all combinations of possible processes. Taking into account that a vast number of manufacturing processes exists and all are described by a high number of parameters, the number of combinations can be very high. Furthermore, the communication effort between the product developer and manufacturer can be quite high, if the product developer has limited knowledge about the technical processes. In this case, a product developer has to consult the manufacturer in order to generate or adjust a bill of processes. Moreover, these process oriented skill description concepts do not represent a suitable basis for an efficient producibility check. The reason for this is that product requirements always have to be converted to a process level in order to compare the requirements with the available resource skills.

Accordingly, in this contribution a description approach is introduced where resource skills are modeled in a product oriented way. This approach follows a strict distinction between product, process and resource information in order to overcome the discussed issues.

3.2 Approaches for the Generation of Adaptation Options and Impact Analysis

In the field of engineering change management and product design, various approaches for the adaptation of products have been addressed. A subset of these approaches deals with the support of the generation of technical solutions (Ouertani et al. 2004). The majority of the works in this field focusses on the modeling of interdependencies between product elements and the analysis of change propagation. These approaches can be roughly classified into two categories, based on the primarily regarded system element. Approaches in the first category focus on *system components* as the focal element of the analyses, whereas the second category puts the focus on *system parameters*.

Approaches of the first category describe interdependencies between system components mostly by probabilities or by binary values. Most of these approaches utilize a matrix-based representation form to model the identified interdependencies. Such approaches have been presented in (Clarkson et al. 2004; Karl and Reinhart 2015; Fei et al. 2011). In sum, these approaches are primarily developed for estimating the potential change propagation of an adaptation on component level. However, regarding the generation of adaptation options, this type of approach is not appropriate. One reason for this is that the approaches are only capable of providing information on which components have to be changed, but cannot specify how to change the component. Furthermore, interdependencies on component level are not sufficiently precise to give accurate information about the adaptation propagation (Hamraz et al. 2013). An adaptation of a component can result in different propagation paths, depending on the component parameter that is supposed to be changed (Reddi and Moon 2009).

The approaches of the second category focus on interdependencies between system parameters. Ollinger and Stahovich (2004) describe interdependencies between product parameters by quantity constraints and causal influences between quantities. The authors use the modeled information for the generation of adaptation options to achieve certain redesign goals in products. Yang and Duan (2012) presented a concept that is based on the work of Ollinger and Stahovich. In their work, interdependencies between product parameters are divided into physical, which cannot be changed, and design links, which are regarded as modifiable. The interdependencies are modeled as algebraic functions. Based on these functions, different change propagation paths are identified for given initial changes. Ahmad et al. (2013) presented a parameter-focused approach which comprises four different layers. The first layer contains requirements that represent changes. These requirements are tracked down to the detail design layer where linkages between parameters are analyzed. Based on this analysis, possible change options are generated.

To sum up, current approaches of the second category provide an appropriate basis for the identification of propagation paths for given adaptation options. Furthermore, they allow the generation of specific adaptation options. However, the main disadvantage of most existing approaches in this category is the quantity and level of detail of the required data. Moreover, the works are rooted in the engineering change or product design community. Thus, the approaches are not directly applicable for manufacturing systems. With regard to the adaptions of manufacturing resources, interrelations between product, process and resource elements have to be considered in the generation of adaptation options.

3.3 Approaches for the Evaluation of Adaptation Options

The cost benefit analysis is a systematic approach to compare different solution options by analyzing the ratio of estimated costs and benefits. In the context of this article, costs include the technical and economical effort to perform an adaptation. The benefits can be measured in terms of flexibility or adaptability. In order to quantify these aspects, different metrics have to be used to define suitable KPIs. The effort estimation of an adaptation can be split up into hardware and software efforts. The following subsections present a selection of approaches for the estimation of efforts and benefits.

3.3.1 Effort estimation for hardware changes

The effort estimation of hardware changes for the adaptation of manufacturing machines has been rarely addressed in the literature. One of the few works that addressed this problem has been presented by Zäh et al. (2011). The authors distinguish between structural and economic KPIs for the estimation of the effort of an adaptation. The structural KPIs consist of factors, such as number of adaptations or number of connections that have to be adapted, whereas the economic KPIs include different cost factors, e.g. labor costs, downtime costs and material costs. A similar approach for the effort estimation has been proposed by Karl and Reinhart (2015). In their approach the authors also consider structural and economic key figures. Here, the four structural key figures: numbers of adaptations, number of interdependencies, number of parts to adapt and reach (share of parts that have to be adapted) have been introduced in order to estimate the effort of an adaption. In addition, three economic key figures are defined to estimate the future value of an adaptation under the consideration of uncertainties. Zhang et al. (2017) presented an approach for evaluating the adaptability of interfaces for the analysis of open architecture products. In their work, interface adaptability is divided into four categories which describe the adaptability from a functional, structural, manufacturing, and operational perspective. Here, the functional adaptability is determined by the feasible inputs and outputs of the interfaces, whereas the structural adaptability is quantified by the degree of structural interface standards. Moreover, the operational and manufacturing adaptability is determined by assembly, disassembly and manufacturing costs of the interfaces. By the use of the proposed metrics, the effort of hardware changes could be determined, as these changes are highly depending of the characteristics of interfaces. However, for the effort estimation of hardware changes it is not sufficient to solely focus on the interfaces. Further factors, such as downtime costs or material costs, also have a high impact on the effort. An interesting approach regarding effort estimation has been presented by Das et al. (2000) in the field of disassembly. The approach introduces a disassembly effort index (DEI) score in order to evaluate the effort of disassembly processes. The DEI score uses seven factors which describe the complexity of a disassembly process with associated scales to evaluate the disassembly effort. Although the approach has been developed for the estimation of disassembly effort, aspects of the approach can be transferred to the estimation of adaptation efforts, as disassembly is a crucial part of an adaptation.

3.3.2 Effort estimation for software changes

Approaches for the effort estimation of software projects can be divided into three major groups: 1) model-based, 2) expert-based and 3) hybrid approaches. COCOMO (Constructive Cost Model) (Boehm 1981) as well as its refinement COCOMO II (Boehm et al. 1995) is an approach of the first category that estimates the effort primarily by the estimation of the number of delivered source instructions. Furthermore, COCOMO takes boundary conditions into account and offers different calculation modes based on the type of software project. Functional Size Measurement approaches also use models to try to derive the effort from the estimated functional size. Since the input data for the model-based approaches can only be vaguely estimated, these approaches lack of accuracy and acceptance.

Expert-based effort estimation is commonly used in practice and is often using analogy. Completed tasks and their known effort are utilized for the determination of the effort for the current task. A comprehensive overview of analogy-based approaches is provided in Idri et al. (2015). Through the consideration of suitable analogy criteria, the characteristics of software adaptation projects can be included in the estimation.

According to ISO 25010, maintainability is a quality characteristic of software. Since the effort for the adaptation of software also depends on its maintainability, software metrics to measure aspects of maintainability (e.g. modularity, analyzability, modifiability) can be applied. In software engineering, different metrics are used to measure these aspects. Well known metrics are Source Lines of Code, Cyclomatic Complexity after McCabe (1976) or the Halstead Metrics (Halstead 1977). Literature review reveals that the metrics used in classical software engineering can also be applied to Programmable Logic Controller (PLC) programs with some additional definitions and language-specific changes (Younis and Frey 2007). Metrics that explicitly focus on PLC programs are for instance presented by Lucas and Tilbury (2005) and Gharieb (2006).

3.3.3 Benefit estimation of adaptation options

The benefit of an adaptation of an automated manufacturing machine can be measured in terms of higher flexibility or adaptability. In literature, flexibility usually is regarded from different aspects. Sethi and Sethi (1990) provide a detailed summary of flexibility research and define eleven kinds of flexibilities in manufacturing. Based on three basic flexibilities (machine, material handling and operation flexibility) the authors derive so called system flexibilities (e.g. routing or product volume flexibility) and aggregate flexibilities. Different methods to measure the proposed kinds of flexibility have been published, e.g. in (Brill and Mandelbaum 1989; Kochikar and Narendran 1992; Das 1996; Buzacott and Mandelbaum 2008; Wahab et al. 2008). However, these approaches do not consider parameter changes in their measurements and thus, are not capable of evaluating flexibility changes on parameter level.

A measure for flexibility and adaptability has been defined in VDI 5201 (2017) (see Fig. 2). The bandwidth of the flexibility for a selected indicator is defined by the boundaries of its current flexibility corridor. Thus, if the bandwidth of the corridor is increased after performing an adaptation, the adaptation made the machine more flexible. The relative change of the flexibility bandwidth after the adaptation can be regarded as the benefit of an adaptation in terms of higher flexibility. This method can be applied on various indicators, e.g. it can be used to illustrate and evaluate the number of product variants or the parameter range of a specific process parameter.

In sum, it can be stated that current approaches are either focusing on the estimation of hardware or software efforts. However, during an adaptation a mechatronic system, both areas need to be evaluated. The approach of VDI 5201 seems appropriate when the flexibilization of parameter ranges is regarded.

4 Proposed Approaches for the Support of the Adaption Process

Suitable approaches for the first three phases (cf. Fig. 1) of the adaptation process are presented in the following three subsections. The proposed approaches address the highlighted challenges of each phase.

4.1 Producibility check of production requests

To provide an efficient producibility check of production requests, this contribution proposes an approach for a product-oriented description of manufacturing resource skills. Here, resource skills are described by product parameters of processable/producible products. In this way, the producibility check can be accomplished by a simple comparison of product parameters.

The approach is based on the established distinction of production domain entities between products, processes and resources (PPR concept) and on the guideline VDI/VDE 3682 (2015). In the PPR concept, products are produced by processes which execute certain changes on the product, whereas a resource represents a hardware and/or software entity that is involved in the execution of a process. The outline of the PPR concept is depicted in Fig. 3, taken from Schleipen and Drath (2009).

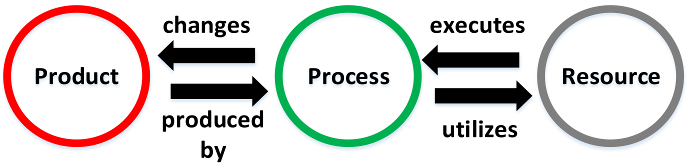


Fig. 3 Outline of PPR concept

The guideline VDI/VDE 3682 specifies a formalized process description for the modeling of processes based on the PPR concept. Here, processes have one or multiple input products which undergo a change during the process and one or multiple output products. For the change of input products, the process utilizes a technical resource. Furthermore, a process can have energy and information as additional input and output elements. An example of the formalized process description is illustrated in Fig. 4. The guideline VDI/VDE 3682 describes technical systems in a similar way to Pahl and Beitz (1997). According to Pahl and Beitz, a technical system performs different functions where each function has energy, material and information signals as inputs and outputs.

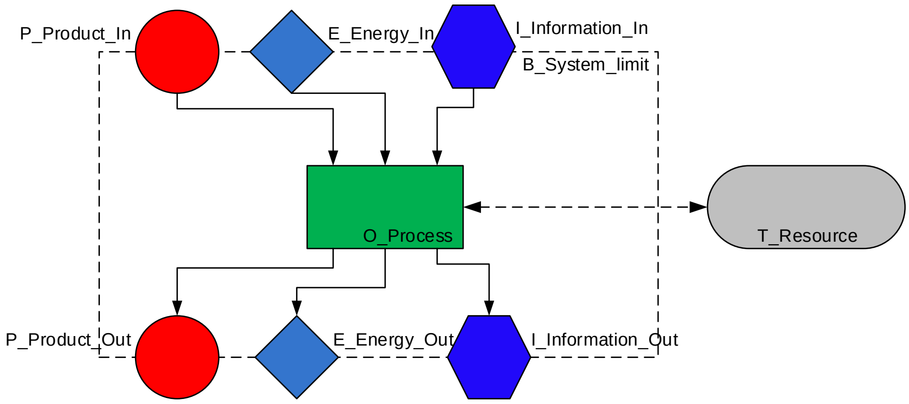


Fig. 4 Example of the formalized process description

According to the PPR concept, resources execute processes in order to change products in a certain way. The proposed approach uses these relations for the description of resource skills in a product oriented way, as the two entities are linked together by the process. Here, resources, processes and products are modeled as parameter spaces. A parameter space consists of different parameters and describes the value range which each parameter can take. Similar to the formalized process description, in this approach, resources are associated with processes that have input and output products. Here, the parameter spaces of these input and output products are utilized to describe the skill of a certain resource. The description concept divides parameter spaces into the following five categories:

- Component-based parameter space: Describes the value ranges of system parameters, i.e. product, process, and resource parameters which are defined by engineering artifacts of the components, without consideration of any interdependencies.
- Structure-based parameter space: Describes the value ranges of system parameters under consideration of structural interdependencies between components, e.g. the spatial relations of components.
- Process-specific parameter space: Describes the value ranges of system parameters under consideration of process-specific interdependencies. For example, the relationship between cutting speed, rotational speed and drill diameter must be taken into consideration in order to execute a drilling process correctly.
- Software-based parameter space: Describes the value ranges of system parameters under consideration of interdependencies that result from the implemented control software.
- Feasible parameter space: Describes the value ranges of system parameters under consideration of all existing interdependencies. Hence, this parameter space represents an intersection of the four previously introduced parameter spaces. Based on this feasible parameter space, manufacturable input and output products are described and thus, also the resource skills.

In order to describe resource skills as product parameter spaces, it is necessary to analyze relationships between a resource, the process and a product. These relationships enable the conversion of resource parameters into process parameters and eventually the conversion into product parameters. These relationships are physical in nature and are either formally described as mathematical expressions or empirically, e.g. in the form of tabular compilations. For example, consider a drill (resource) that has the parameter "flute length" with a value of 18 mm. The corresponding drilling process has the parameter "drilling depth". Since the drill cannot manufacture a hole which is deeper than its flute length, the relationship between these two parameters can be described by the expression: drilling depth ≤ flute length. Accordingly, the drilling depth can be in a range between 0 mm and 18 mm. The process parameter "drilling depth" can be converted into the output product feature "hole depth". Here the relationship between these two features is described by the equation: drilling depth = hole depth. Thus, based on these relations and value ranges, the resource is capable of manufacturing an output product which has a hole with a depth between 0 mm and 18 mm. It is important to note that in the example only the component-based parameter space is described. Accordingly, the value ranges of the output product can be limited by interdependencies of the other parameter spaces. An overview of the description concept is depicted in Fig. 5. Based on this skill description concept, a production request only needs to be described by the desired input and output product without specifying process specific information. The producibility check can then be accomplished by only comparing the parameters of the requested products. Furthermore, this description facilitates the generation of adaptation options. If a production request is not producible, the analysis of the parameter spaces gives an indication how a limitation can be solved, e.g. by a hardware or software adaptation (see subsection 4.2.2).

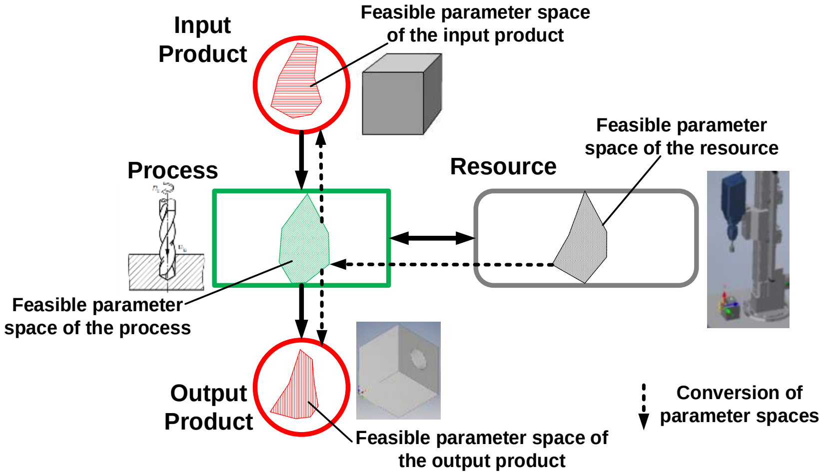


Fig. 5 Overview of the skill description concept

For the producibility check, the requirements of a manufacturing request are modeled with the formalized process description on a detailed level. Based on this model, it can be analyzed which resource is capable or not capable of fulfilling the request by using the described skill description concept for manufacturing machines.

4.2 Generation of adaptation options

4.2.1 Model of interdependencies between manufacturing system elements

The modeling concept that is utilized in this contribution is based on the information model presented by Hoang et al. (2017a). Accordingly, interdependencies are considered between three types of elements, namely: products, processes and resources. All elements are specified by parameters. In the concept, interdependencies are modeled qualitatively as constraints and correlation relations. The interdependency information is stored in form of a Multiple-Domain Matrix (MDM). The MDM is a compact and simple description method which is also frequently applied for modeling interdependencies in complex products (Keller et al. 2005). Thus, this description method is also suitable for the modeling of interdependencies in manufacturing systems. An example of such a MDM is illustrated in Fig. 6. A brief application example of the matrix is given in section 4.2.2.

In the MDM, two different types of interdependencies are used to describe the relations between parameters. The first type describes correlation relations using the symbols “+”, “-“ and “0” (see Fig. 6). The second interdependency type describes constraints and is denoted by the symbols ▲, ▼, and ● (see Fig. 6). The constraints state if a change of a parameter value is limited by another parameter value. The first symbol denotes that a constraint for the increase of a parameter exists. Accordingly, this symbol represents a lesser than and lesser or equal to constraint. The second symbol denotes a constraint for the decrease of parameter and represents a greater than and greater or equal to constraint. The last symbol represents an equal constraint. The two interdependency types can also be used in combination. The combination then denotes if a parameter change is limited and how the limitation can be relaxed. For instance, in Fig. 6, the increase of resource parameter 2 is limited by the value of resource parameter 3. This limitation can be relaxed by increasing the value of resource parameter 3 as there is a positive correlation.

In the matrix, the rows marked in green describe which relations and constraints have to be considered if a process parameter (PP) is adapted. Here, the correlation relations denote which process parameters or resource parameters can be changed in order to adapt a specific process parameter. As an adaption of a process parameter can cause unintentional incompatibilities with product parameters (ProdP), the adaption is constrained by product parameters. These constraints ensure that incompatibilities regarding the manufacturing of a product are avoided.

	Process_Parameter_1	Process_Parameter_2	Process_Parameter_3	Process_Parameter_4	Resource_Parameter_1	Resource_Parameter_2	Resource_Parameter_3	Resource_Parameter_4	Product_Parameter_1	Product_Parameter_2	Product_Parameter_3
Process_Parameter_1			▲+	+	0		-			▲+	
Process_Parameter_2						+		-	▼+		
Process_Parameter_3						-		+			●+
Process_Parameter_4	+				+		-		▼+		
Resource_Parameter_1	0			+		▼-					
Resource_Parameter_1	0			+			▲-				
Resource_Parameter_2		+	-				▲+				
Resource_Parameter_2		+	-					▼-			
Resource_Parameter_3	-			-	●+			●+			
Resource_Parameter_4		-	+			▼-					
Product_Parameter_1		▲+									
Product_Parameter_2	▼+										
Product_Parameter_3			●+								

Symbology

+	: Positive correlation relation between two parameters
-	: Negative correlation relation between two parameters
0	: Interdependency between two parameters where no statement about the correlation can be made
▲	: Constraint for the increase of a parameter
▼	: Constraint for the reduction of a parameter
●	: Equal constraint

Fig. 6 Example of the MDM information basis

The adaption of resource parameters can be constrained by other resource parameters and can also have an impact on process parameters which were not supposed to be changed. The information, which is stored in the grey colored rows of the MDM, describes these kinds of resource parameter interdependencies. Interdependencies between resource parameters are differentiated into three categories, namely: structure-related, software-related, and process-related interdependencies. These categories are connected to the parameter spaces introduced in section 4.1. Based on the categories, it can be determined what type of adaptation, e.g. hardware or software change, has to be applied in order to overcome or relax the constraints.

Interdependencies depicted in the red colored rows can be used for the impact analysis of product parameter changes. By analyzing this matrix area, it can be concluded whether a product parameter change may require a process parameter change.

4.2.2 Impact analysis and adaptation option generation

The proposed approach is based on the notion that certain changes of product parameters result in non-feasible requirements, because of an insufficient range of parameters that is provided by a manufacturing resource (cf. section 2 – characteristic C2). In this approach a directed rooted tree graph is used for the visualization and determination of adaption options and their impact. Rooted tree graphs are regarded as a suitable visualization method which supports the generation of adaptation options (Keller et al. 2005). The tree graph consists of nodes which are connected by edges. Here, nodes represent system parameters and the edges denote the interdependencies between the parameters. A rooted tree graph has exactly one root node which represents the origin of the tree. Other nodes in the graph are child nodes of the root node. Each step from the root node to a bottom child node is called a level. The level of the root node is ‘0’ and is increased by one at each level. In the rooted tree, correlation relations and equal constraints are denoted by continuous edges. The other two types of constraints are depicted as dashed edges. Additionally, edges depict information regarding the type of correlation.

The approach for the option generation and impact analysis utilizes the information stored in the MDM to generate a rooted graph. Based on the constructed tree graph, adaptation options are derived. The approach is based on the concept proposed by Hoang et al. (2017b). Step 1 of the option generation is the analysis of the red colored matrix rows. Here, it can be analyzed which process parameters have to be adapted as a result of the given product parameter changes. Each of the identified process parameters represents a root node of a tree graph. Additionally, the type of change (“increase” or “decrease”) can be determined for parameters with values that are described by an ordinal, interval or ratio scale. For parameters with nominal values, a type of change cannot be determined. The type of change is depicted in the tree by arrows (see Fig. 7).

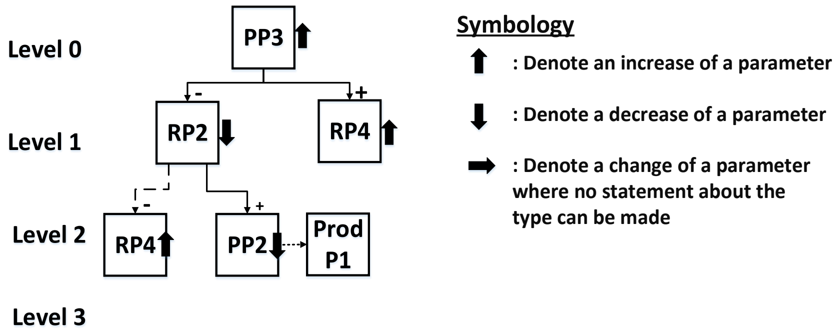


Fig. 7 Rooted tree graph for the increase of process parameter 3 (level 0 -2)

In step 2, the green rows of the MDM are analyzed to determine resource parameters that have a direct influence on the root node including the required type of change. These resource parameters represent the level 1 child nodes of the rooted tree. Additionally, it is analyzed if the change of the root node causes incompatibilities with other product parameters. In step 3, the grey colored MDM rows are analyzed with the aim to identify resource parameters that constrain the adaptation of resource parameters in the preceding level. In the same step, it is examined whether process parameters that are not supposed to be adapted are affected by the change of resource parameters in the preceding tree level. In case of an unintentional change of a process parameter, it also needs to be analyzed if this change is constrained by a product parameter. This is accomplished by the information stored in the green colored MDM rows. Based on the three constructed tree levels, potential adaptation options can be generated in step 4. For this purpose, tree paths have to be identified which start at the root node and contain resource parameter nodes of succeeding tree levels. In addition, the last resource parameter node in the path must fulfill one of the following criteria: a) the last path node has resource parameter child nodes and is not located in the last level of the constructed tree, b) the last path node is only connected by dashed edges to its resource parameter child nodes, c) the last path node is in the last level of the full tree.

Such a path including the associated process and product parameter child nodes of the contained resource parameter nodes represents a potential adaptation option. Note here that potential options can be combined to generate further potential options. Here, it is required that the corresponding paths in the combination are no sub paths of one another. Furthermore, it has to be analyzed whether a combination is free from contradictions (i.e. no increase and decrease of a parameter at the same time). To check if a potential option represents a valid adaptation option, quantitative analyses are conducted in step 5. Here, for paths which fulfill criterion b) it needs to be analyzed if the constraints of the last path node to its child nodes are not violated by the change of its parent node. If the change does violate a constraint, the corresponding path is not a valid adaptation option. Furthermore, for all potential options it has to be examined whether the changes of resource parameters in the option result in undesired/unpermitted changes of process and/or product parameters. In this case, these paths represent invalid adaption options. If at the end of step 5 no valid options or an insufficient amount of valid options have been generated, the tree graph can be iteratively enhanced by additional levels as depicted in Fig. 8.

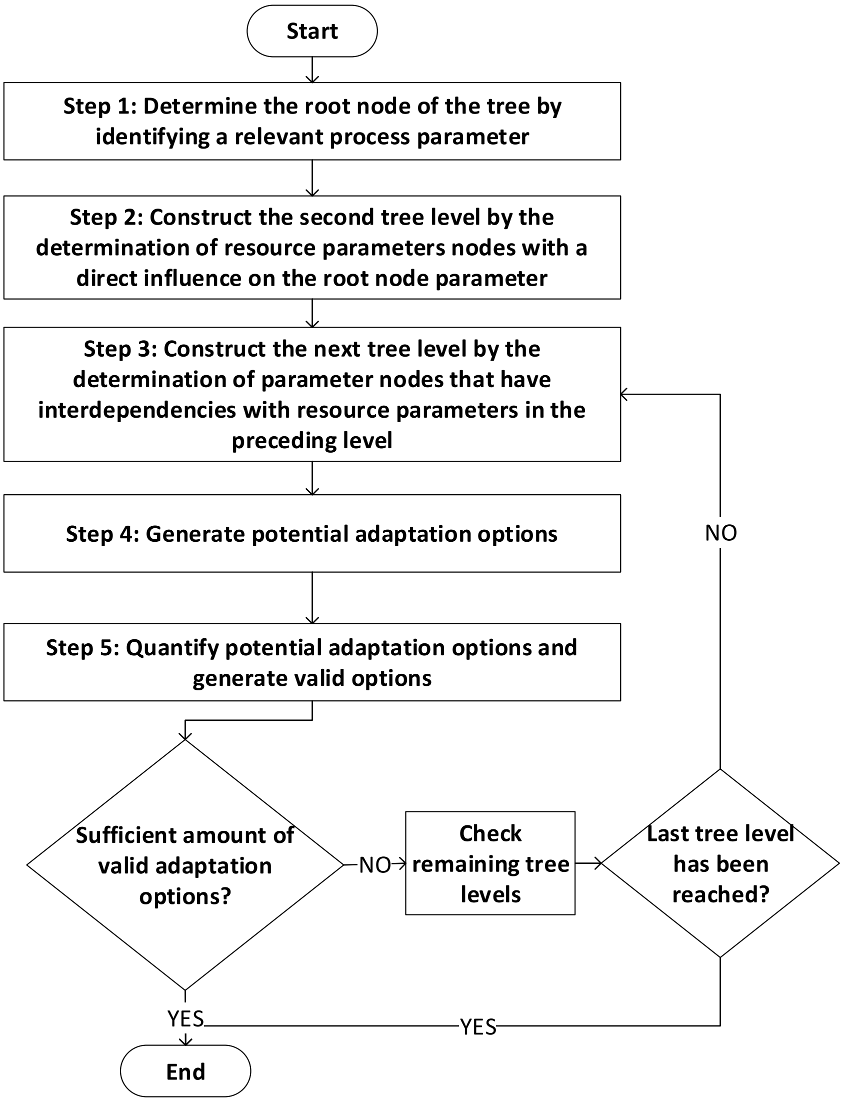


Fig. 8 Overview of the approach for the generation of adaptation options

In the following, the approach is illustrated based on the MDM example depicted in Fig. 6. Assume that a product developer designed a new product variant where ProdP3 is increased. Based on the MDM, it can be analyzed that this increase also requires an increase of PP3 in order to produce the new variant. Thus, the root node of the tree graph represents the increase of PP3. In step 2, it can be identified that RP2 and RP4 have a direct influence on PP3. Here, RP2 can be decreased (negative correlation) and/or RP4 can be increased (positive correlation) to accomplish the desired increase of PP3. The analysis in step 3 reveals that the increase of RP4 is not constrained and has no further impact on other parameters. Moreover, the analysis shows that the decrease of RP2 is limited by RP4 and results in a reduction of PP2, which is limited by ProdP1. Based on these analysis results, the first three levels of the tree graph can be constructed (see Fig. 7).

In step 4, three potential adaptation options can be determined by the identification of outlined paths in the constructed tree and path combinations.

The three potential adaptation options and the quantitative analyses in step 5 of the generated potential options are depicted in Fig. 9. It can be seen that PP3 is supposed to be increased by 4 units. Therefore, RP4 needs to be increased by 2 units in adaptation option 1, as the positive correlation relation between PP3 and RP4 has the factor 2. In adaptation option 2, RP2 has to be decreased by 4 units. This decrease results in a decrease of PP2 by 8 units. However, the decrease of PP2 is limited by ProdP1. As the constraint allows a decrease of 10 units (depicted in green) of PP2, the constraint is not violated in this case. The decrease of RP2 is constrained by a resource parameter node in the succeeding level. In this case, the decrease of RP2 by 4 units violates the constraint imposed by RP4, as only the decrease of 1 RP2 unit (depicted in red) is allowed. Hence, the potential adaption option 2 is not feasible. Adaption option 3 comprises of the increase of RP4 by 1 unit and the decrease of RP2 by 2 units which, in sum, accomplish the desired increase of PP3 by 4 units. Here, the decrease of RP2 also leads to a reduction of PP2 which does not violate the constraint imposed by ProdP1. However, here, the increase of RP4 by 1 unit relaxes the constraint between RP2 and RP4 by 1 unit because of the negative correlation. As a result, RP2 is allowed to decrease by 2 units (depicted in green) without violating the constraint. Thus, this option represents a valid adaptation.

As the generated adaptation options are on a parameter level, it has to be analyzed whether the included parameter changes can be accomplished by hardware or software adaptations based on the parameter space categories (cf. section 4.1). As a result, a list of hardware and software components is generated that can be utilized for the effort estimation of the adaptation options.

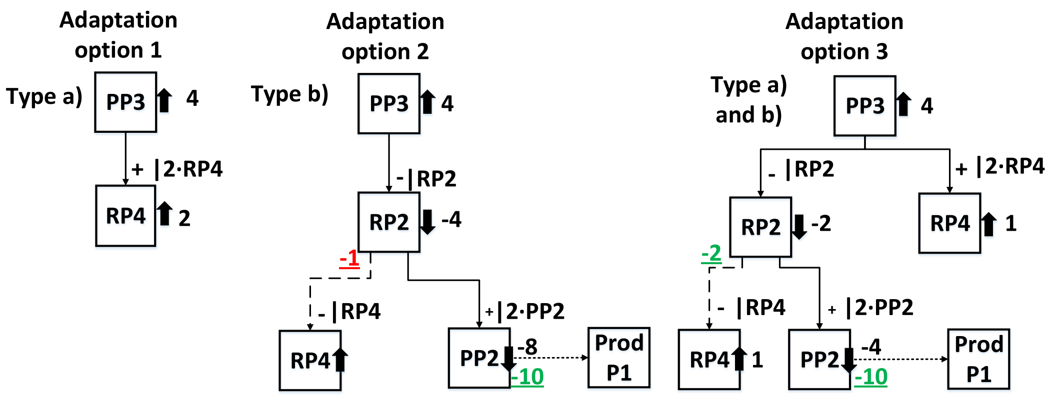


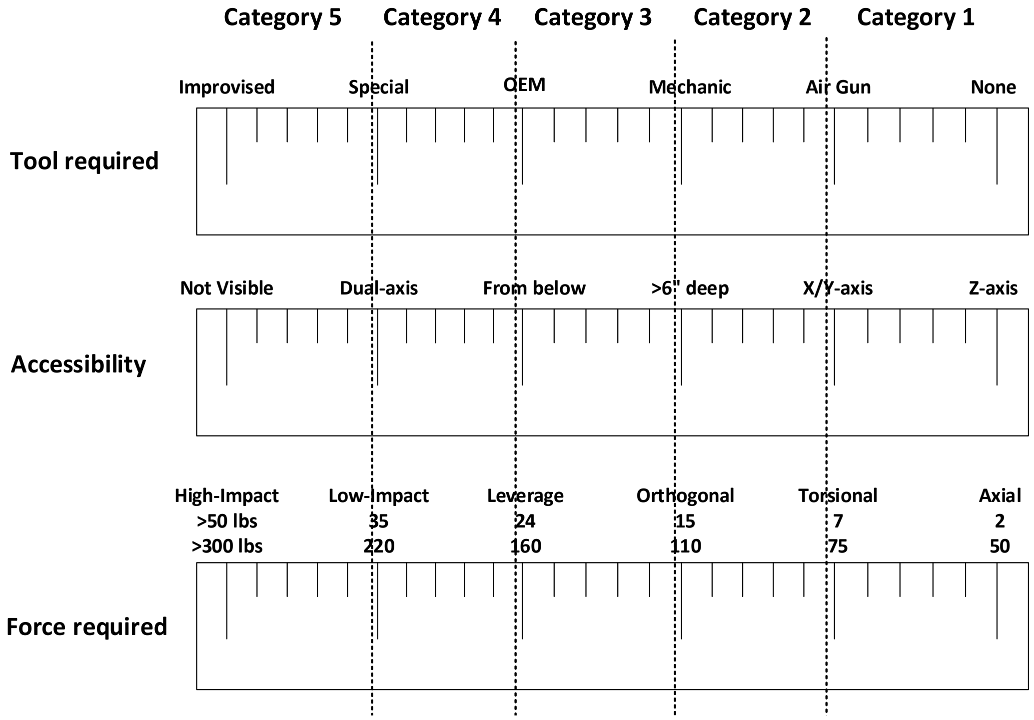
Fig. 9 Quantitative analysis of potential adaptation options

4.3 Concept for the Estimation of Efforts and Benefits of Adaptation Options

The following subsections present the concepts for the estimation of efforts, separated by hardware and software efforts and for determining the flexibility of a proposed adaptation option.

4.3.1 Effort estimation of hardware adaptations

In this subsection, the concept for the estimation of effort regarding hardware adaptations is presented. The concept is based on a literature review of 21 references regarding the effort estimation of changes in the hardware structure of different technical systems. The aim of the proposed concept is to provide an efficient way to estimate the effort of hardware adaptations with respect to the ratio of accuracy and required data. In the developed concept two categories of effort factors are considered, namely: economic factors and structural/technical factors. The first category consists of the factors *downtime costs*, *labor costs*, and *adaptation time*. In order to allow a better comparison between the economic efforts of adaptation options, the ratio between the downtime and labor costs to the adaptation time is computed and used for the comparison of options. The second category consists of the two quantitative factors *number of components* that ought to be adapted and *number of interfaces* that have to be adapted, due to the adaptation of a component, as well as the three qualitative factors *tools required*, *accessibility*, and *force required*. The qualitative factors are described by a scale with five categories based on the scales presented in Das et al. (2000) (see Fig. 10 - top). Here, the first category represents the lowest and the fifth category the highest effort. For a detailed description of the scale the reader is referred to Das et al. (2000). This scaling system represents an efficient and effective way to measure the disassembly effort, and is widely accepted in the field of disassembly of technical products (Harivardhini and Chakrabarti 2016). Since the effort of hardware adaptations of manufacturing machines highly depends on the disassembly of components and interfaces, the authors regard this scale as also suitable for the evaluation of adaptation efforts. The three qualitative factors are used to evaluate required effort for change of each component and interface. The evaluation is supposed to be accomplished manually. The results of the evaluation of the structural/technical factors are represented in scoring cards (see Fig. 10 - bottom). In the given example, seven components are supposed to be adapted and their adaptations require a medium up to high level of effort, as the majority of the components are ranked in the categories 3 to 5 regarding the qualitative factors.



	Category 5	Category 4	Category 3	Category 2	Category 1	∑Components
Tools required	2	2	3	-	-	
Accessibility	1	1	1	2	2	7
Force required	1	3	2	1	-	
∑Components	4	6	6	3	2	

Fig. 10 Categories of the qualitative factors (Das et al. 2000) and example of a scoring card

4.3.2 Software effort estimation based on analogy

To estimate the effort for the changes in the control software that are required for a specific adaptation option, a methodology that combines case-based reasoning with the usage of software metrics has been developed. This allows the estimation of effort based on the experience of previous adaptations. Case-based reasoning is an analogy based method that derives solutions from a repository that can be adapted and revised for the current task. In the case of effort estimation, the repository contains a set of finished tasks, their properties and their measured effort. By retrieving similar tasks from the repository, an estimation of the effort can be supported.

In order to apply a case-based reasoning system for the effort estimation, suitable properties for the task and the object to be changed, i.e. the control software, need to be defined. Fig. 11 depicts the effort estimation process using the described approach.

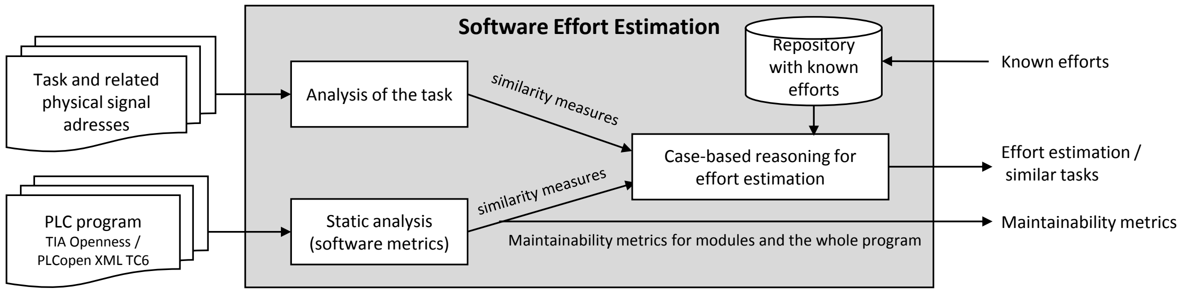


Fig. 11 Effort estimation for software changes in PLC programs using case-based reasoning

Based on the proposed adaptation options that have been generated in the previous step, a specific type of software change can be assigned to the adaptation option. For instance, the proposed actions can be classified into different categories like adding or removing module interfaces, changing the functionality of a module etc. Furthermore, the physical addresses of the signals that are interrelated with a resource are known and can be provided to the software effort estimation system. To determine the parts of the software that are affected by the change of the technical process, static analysis can be used to identify the relevant parts of the software that are associated with the technical resources to be changed. Technical resources in this context are sensors that provide input information to the controller and actuators that are controlled via the outputs of the PLC/controller.

To calculate measures for the similarity of PLC programs and especially for the affected parts of the program (function blocks, functions), the current PLC program is statically analyzed in a XML format in order to apply software metrics that quantify different maintainability aspects. The calculated metrics serve as a similarity measure on the one hand but can also give the user additional hints about the maintainability of the software, e.g. whether the software is properly modularized or that the software is barely commented.

An overall similarity between two cases can be calculated by a weighted average of the previously normalized similarity measures. The normalization is necessary because the values of some of the software metrics are not limited, e.g. source lines of code or cyclomatic complexity. Effort can be estimated by adopting and adapting the effort of a case that has been determined as similar. Further information on the approach and details on the realization can be found in Marks and Weyrich (2017).

4.3.3 Benefit estimation by comparing flexibility corridors

The proposed approach for determining the benefit that is gained by the implementation of a previously generated adaptation option is based on the flexibility corridors described in guideline VDI 5201 (2017). Since each adaptation option in the context of this article aims at changing process parameters by changing the corresponding parameter(s) of resources, the relative change of the bandwidth of the process parameter can be regarded as a measure for the benefit of a specific adaptation option (see subsection 3.3.3). If an adaptation option changes multiple process parameters, the relative changes of the bandwidth of parameters are clustered according to the flexibilities introduced by Sethi and Sethi (1990). Based on these clusters, it can be determined which kinds of flexibility of the manufacturing machine have changed. Therefore, the sum of the relative bandwidth changes in a cluster is computed. Note here that a sum of the relative bandwidth change can also be zero, as the flexibilization of one process parameter can result in decrease of the flexibility bandwidth of a different parameter in the same cluster. An adaptation can also only aim for shifting the level of the flexibility corridor without changing its bandwidth. In such a case, the machine can operate on a different corridor level, but the flexibility of the regarded parameter has not been changed.

5 Agent Architecture for the implementation of a Decision Support System

To implement the proposed assistance concept for the adaptation process, the paradigm of software agents was used. Software agents have been subject to research for more than 20 years and applications can be found in various fields (e.g. Frayret et al. 2007; Bussmann et al. 2010; Leitao 2015). Although there is no commonly agreed definition of an agent, a consensus exists regarding the basic properties of agents: reaction to the environment, autonomy and goal-orientation (Leitao 2015; Wooldridge and Jennings 1995). Another property is the ability of agents to collaborate on a task in order to achieve an overall goal. The collaboration of agents takes place in a Multi Agent System (MAS). Through their inherent properties, software agents are qualified to be used in scenarios where flexibility and the assessment of exploratory solutions are required (Leitao et al. 2013). Since these aspects are also valid for the adaptation process of automated manufacturing machines, the agent paradigm is suitable for the implementation of the previously described assistance concept in section 4.

The agent paradigm allows to render the whole process transparent to the user because the goal-oriented view corresponds to the human way of thinking. Transparency of the process and the result takes an important role when it comes to the user acceptance of a decision support system (Woods 1986). The architecture of the MAS has been developed using the Gaia methodology (Wooldridge et al. 2000), a method for agent-oriented software development. Gaia offers a methodology for the analysis and design phase known from traditional software projects that is extended and combined with aspects that arise from the use of the agent paradigm. Gaia has been proven to be suitable for designing MAS architectures. The results of the application of the Gaia methodology are described in the following subsections.

5.1 Identification of roles in the multi agent system

The roles in the multi agent system have been defined in accordance to the modus operandi of human experts in the adaptation process. The roles have been divided into three types depending on their type of knowledge or their pursued goal. The first type of roles contains information about the manufacturing machine, its topology, the technical process and domain specific knowledge (Type A). Roles of this type share their knowledge upon request with other roles. Roles of Type B are responsible for analyzing production requests and for generating and evaluating adaptation options by different criteria. These roles include the algorithms presented in Section 4. Roles of the third type (Type C) support the final decision making process by negotiation. A slightly simplified overview of the identified roles and their short description is given in Table 2. Since solely the evaluation of the aspects *flexibility* and *effort* was discussed in section 4.3, only these aspects are considered by the “Analyzer” and “Debater” roles in this article. However, the architecture allows the integration of other aspects like *adaptability* or *modularity* by integrating the corresponding roles in the multi agent system.

Table 2 Overview of the identified agent roles according to (based on Marks et al. (2017))

Role	Description	Type
MachineTopologyProvider	Holds and provides information about the topology of the machine. For clarity, the machine can be divided into different functional entities.	A
FunctionalEntityRepresentator	Holds the knowledge and information model of a functional entity.	
SoftwareAdministrator	Holds information about the software of one controller.	
ProductRequestAnalyzer	Assesses the producibility of a given production request.	B
AdaptationOption	Holds information about one possible adaptation option and assesses its own impact on other system elements.	
EffortAnalyzer	Analyzes the effort of a given adaptation option.	
FlexibilityAnalyzer	Analyzes the flexibility of a given adaptation option.	
EffortDebater	Tries to select an adaptation option with the least effort.	C
FlexibilityDebater	Tries to select an adaptation option with the highest flexibility.	

The modular structure of an automated manufacturing machine on hardware level can usually be determined by identifying the manufacturing and handling processes that are performed by the machine. For instance, a conveyor belt transports workpieces whereas a drilling station changes the properties of the workpiece. Unfortunately, the structure of the software controlling the technical process usually does not correspond to the modular structure of the hardware in most legacy systems. Commonly, PLCs are used for controlling existing manufacturing machines. The control software might be distributed on several controllers.

In order to take that different views on modularity into account, the separated roles *SoftwareAdministrator* and *FunctionalEntityRepresentator* are used to model the information about the software and the controlled hardware. Physical modules are represented by the role *FunctionalEntityRepresentator* which contains information about the related resources and the process that is performed by this functional entity (FE). The role *SoftwareAdministrator* contains information about the software that is running on one controller. Information about the topology of the machine, i.e. the feasible sequence of the functional entities, is provided by the role *MachineTopologyProvider*.

Since specific parts of the software control the process of a functional entity, linkages between the roles *SoftwareAdministrator* and *FunctionalEntityRepresentator* have to be considered. These links can be established by analyzing the signals (sensor inputs and actuator outputs) between controller and technical process. Possible applications that can make use of these links are effort estimation of software changes that are related to changes of the hardware and the identification of process parameters that are set to a specific value in the software. For instance, if the process speed is limited by the software for quality purposes (limitation by the software-based parameter space), it would be possible to adjust this parameter and to increase the output quantity. Fig. 12 displays the interactions of the identified roles during the adaptation process.

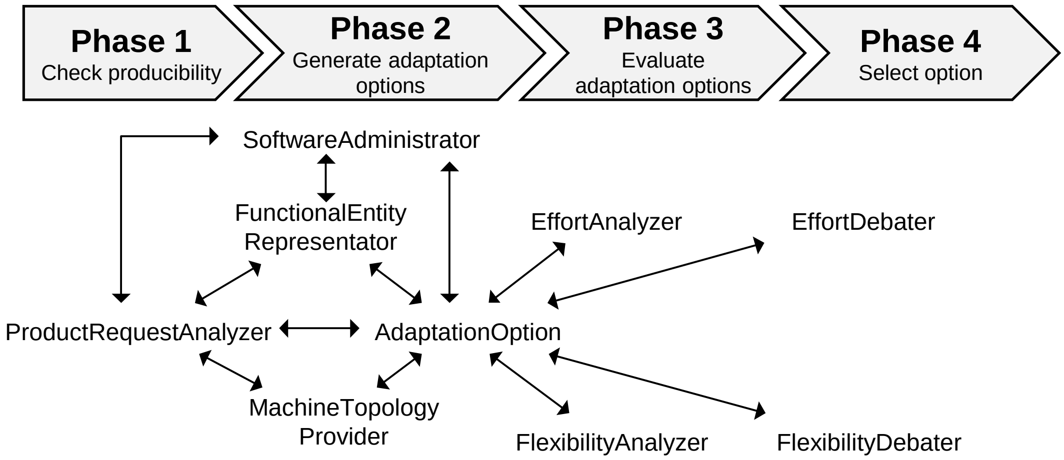


Fig. 12 Graphical interaction model of the roles in the corresponding phases of the adaptation process

In the first phase, the producibility of the user-defined production request has to be assessed. Therefore, the role *ProductRequestAnalyzer* processes the user input in form of a formalized process description and requests information about the manufacturing machine by the roles *MachineTopolgyProvider*, *SoftwareAdministrator* and *FunctionalEntityRepresentator*. Using this information, it is assessed whether the available production resources and processes and their parameter space fit to the production request according to section 4.1. During this analysis, the role *ProductRequestAnalyzer* identifies incapabilities for the given production request.

In the second phase of the adaptation process, suitable adaptation options need to be generated using the approach and algorithm described in section 4.2. The necessary information for this step is distributed over the roles *FunctionalEntityRepresentator* (containing the MDM for the functional entity) and *SoftwareAdministrator* (containing the software model). The impact of an adaption option on other system elements is analyzed by the role *AdaptationOption*.

In the third phase of the adaptation process, the generated solutions are analyzed in terms of effort and flexibility by the corresponding analyzer roles according to section 4.3. Incomplete or missing information is requested from the user in order to perform the analysis and evaluation. The output of these analyses is given as KPIs that can be used in the following decision phase.

Although the decision-making process in the fourth phase is not in the scope of this article, the aim of the roles in this phase is briefly depicted. Based on their aspect, the “Debater” roles interact with each other and negotiate a preselection of the previously generated adaptation options that shall be displayed to the user. The user is then capable of selecting the most appropriate proposal for the given boundary conditions and strategical considerations.

5.2 Mapping of roles to Agent Types and Agent Interaction

Following the Gaia methodology, the previously identified roles are assigned to agent types in the subsequent design phase. In many cases, this might be a one-to-one relationship; however, it is also possible to assign multiple roles to an agent for efficiency reasons or convenience. Agents are also able to represent multiple roles in different phases of the agent system. Therefore, the “Analyzer” and “Debater” role of the regarded aspects are combined into on AspectAgent since their regarded aspect is similar. One aspect agent for each considered aspect, i.e. flexibility or effort, will be initiated. Fig. 13 shows the identified agent types and the number of their instances in the multi agent system.

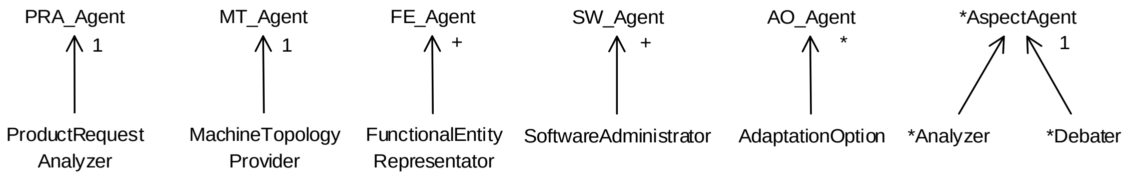


Fig. 13 Agent model of the decision support system according to Marks et al. (2017)

The role *ProductRequestAnalyzer* is assigned to the PRA_Agent which is instantiated once in the MAS. The PRA_Agent provides a graphical user interface that allows to define a production request that shall be analyzed by the MAS. In the first phase of the adaptation process, the PRA_Agent acts as a coordinator between other agents and requests information to determine the producibility of the production request.

Each functional entity is represented by a FE_Agent. In most cases, adaptations within a functional entity can be regarded as independent from other functional entities as long as interfaces between them are not affected. Therefore, the efficiency of the system design can be increased because most adaptation options only have to be analyzed within a limit space, namely one FE_Agent. Additional checks with other FE_Agents can be limited to the affected interfaces between them, e.g. product parameters or material handling interfaces. For instance, changing the drilling diameter will most likely not affect other processes with the exception of quality checks of the previously drilled hole. The FE_Agents includes information about parameter spaces and interdependencies between product, process and resource parameters.

The roles *MachineTopologyProvider* and *SoftwareAdministrator* are each represented by an agent who encapsulates the knowledge of the corresponding role. Both agents have relations to the FE_Agents based on the interdependencies described before. The MT_Agent contains feasible sequences of the processes that are offered by the functional entities. The SW_Agent includes a model of the control software and communicates with the FE_Agents in order to find out which elements of the software code are related to which functional entity. In the case that multiple controllers (PLCs) are used to control the manufacturing machine, one instance of the SW_Agent will be instantiated for each controller by the agent system.

After the initialization of the system, a production request can be entered by the user. In case the production request cannot be fulfilled by the current manufacturing machine, the PRA_Agent analyzes the incapabilities and instantiates one AdaptationOptionAgent (AO_Agent) for each possible adaptation option that has been determined in the second phase of the adaptation process. Each AO_Agent analyzes the impact of the proposed adaptation option on other system elements, i.e. FE_Agents and the SW_Agent(s). By analyzing the parameter spaces introduced in section 4.1 it can be found out whether an adaptation option can be implemented solely by changes of the software or whether a change in the hardware is needed, which may also require a subsequent change of the software. The investigation of the change propagation can be used by the EffortAspectAgent in the subsequent phase where it tries to analyze the effort needed for the implementation of this adaptation option. Additional information can be requested from the user if necessary. The flexibility of each adaptation option is assessed by the FlexibilityAspectAgent.

Then the negotiation process starts that aims at proposing a small set of the most suitable adaptation options to the user. This is necessary in order to decrease the number of possible adaptation options that might be too high for the user to take all options into account before making his decision under strategic considerations.

6 Validation of the Concept using a Lab-size Demonstrator

The proposed concept for the support of the adaptation of manufacturing machines has been applied to a modular production system (MPS) by Festo Didactic at the University of Stuttgart. The MPS serves as an exemplary manufacturing machine to evaluate the applicability of the proposed concepts and the implementation of the agent architecture. The multi agent system has been implemented using the Java Agent Development Framework (JADE) which provides a middleware for the convenient and efficient implementation of multi agent systems.

6.1 Description of the Modular Production System

The MPS consists of different modules that perform manufacturing and material handling processes. Firstly, the workpiece is pushed out of a stack magazine and transported to a testing station by a vacuum gripper. After checking color and material, the workpiece is either sorted out or transported to a rotary indexing table using a conveyor belt. On the rotary indexing table, four positions exist: workpiece input, drilling, check drill hole and output. At the output position, the workpiece is picked up using a vacuum gripper mounted to a portal crane and sorted into three stack magazines or onto a slide for not properly manufactured goods. The system is able to process three different types of workpieces that can be differentiated by their color and material. The manufacturing process is controlled by a PLC.

In the first step, the interdependencies between product, process and resource parameters have been investigated and modeled. In sum, 131 parameters have been identified (6 product parameters, 41 process parameters and 74 resource parameters) to describe the properties of the MPS. The identified functional entities are: 1) Distribution (Fig. 14), 2) Workpiece checking and transportation, 3) Workpiece processing and 4) Sorting and Storing. The division into functional entities allows the creation of four (smaller) MDMs that are only linked with each other by the restrictions of the product parameters since interdependencies between process and resource parameters only occur within a module. The four MDMs contain 133 interdependencies and restrictions. The matrix for the distribution module is depicted in Fig. 15. Furthermore, the current parameter ranges of the process and resource parameters have been determined. This information is used as the model of the manufacturing machine within the multi agent system.

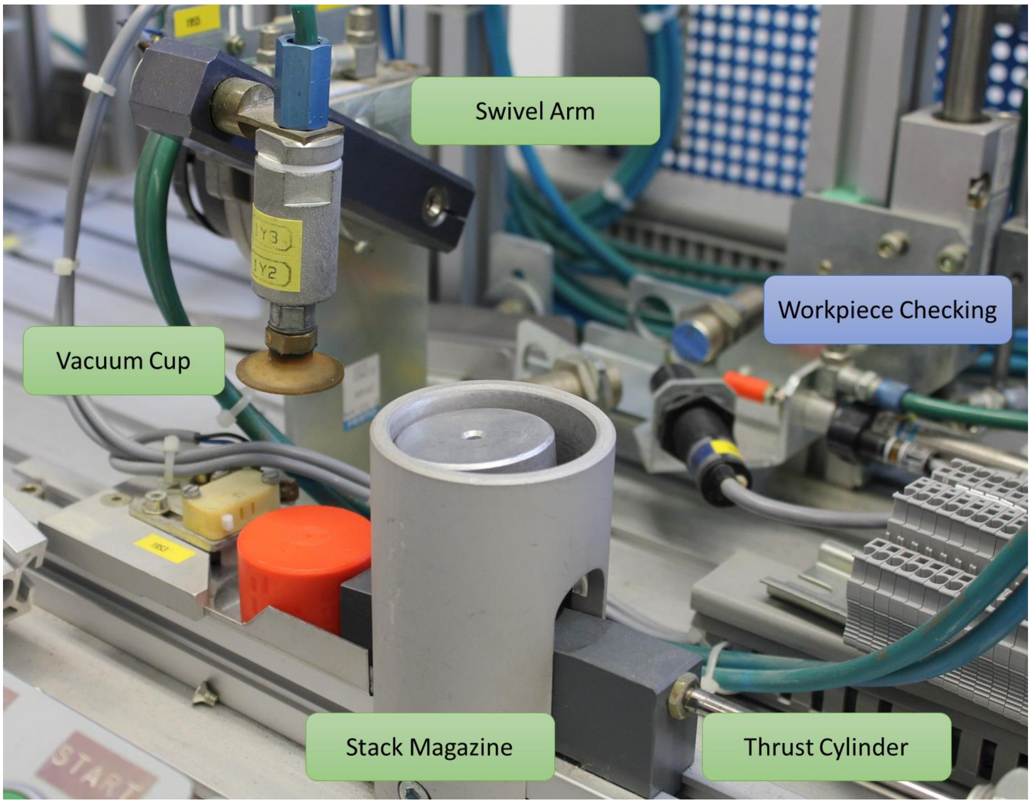


Fig. 14 Overview and components of functional entity “Distribution”

Distribution	WP-Diameter_Max	WP-Diameter_Min	WP-Height_Max	Pushout_Speed	Pushout_Distance	PickAndPlaceUnit_Transport_Distance_y	PickAndPlaceUnit_Transport_Distance_z	PickAndPlaceUnit_ProcessSpeed	WP-Weight_Max	StackMagazine_Diameter	StackMagazine_OpeningHeight	Distance_StackMagazine_PickAndPlace	ThrustCylinder_PushingForce	ThrustCylinder_PushingSpeed	ThrustCylinder_OpPressure	ThrustCylinder_PistonStroke	ThrustCylinder_ValvePressure	SwivelDrive_OpPressure	SwivelDrive_Speed	SwivelArm_Length	VacuumCup_OpPressure	VacuumCup_Diameter	VacuumCup_Force	VacuumGenerator_Vacuum	Product_Diameter	Product_Height	Product_Weight
WP-Diameter_Max										+															▼		
WP-Diameter_Min										+												+			▲		
WP-Height_Max											+									-						▼	
Pushout_Speed														+													
Pushout_Distance												+				+											
PickAndPlaceUnit_Transport_Distance_y																				+							
PickAndPlaceUnit_Transport_Distance_z																				+							
PickAndPlaceUnit_ProcessSpeed																			+								
WP-Weight_Max													+														▼
WP-Weight_Max																							+				▼
StackMagazine_Diameter	+	+																									
StackMagazine_OpeningHeight			+																								
Distance_StackMagazine_PickAndPlace					+											▲+											
ThrustCylinder_PushingForce									+						●+												
ThrustCylinder_PushingSpeed				+											●+												
ThrustCylinder_OpPressure													●+														
ThrustCylinder_OpPressure														●+													
ThrustCylinder_OpPressure																	●+										
ThrustCylinder_PistonStroke					+							▼+															
ThrustCylinder_ValvePressure															●+												
SwivelDrive_PermissibleLoad									+																		
SwivelDrive_OpPressure																			●+								
SwivelDrive_OpPressure																					●+						
SwivelDrive_Speed								+										●+									
SwivelArm_Length			-			+	+																				
VacuumCup_OpPressure																		●+									
VacuumCup_Diameter		+																					●+				
VacuumCup_Force									+															●+			
VacuumCup_Force									+													●+					
VacuumGenerator_Vacuum																							●+				
Product_Diameter	▲+	▼+																									
Product_Height			▲+																								
Product_Weight									▲+																		

Fig. 15 Multiple domain matrix with interdependencies for the module "Distribution"

6.2 Description of the change scenario

To demonstrate the concept and the behavior of the agent system, a small scenario is described in the following. Due to changes in customer demand, the diameter of the round workpieces shall be increased from 42 mm to 50 mm. Since the material of the workpieces does not change, the weight of the workpieces increases from 500 g to 600 g at the same time. The sequence of the production process shall not be changed. This information is entered in the graphical user interface of the multi agent system. An additional CoordinatorAgent provides the GUI and interacts with the agent system described in section 5. After clicking the “Start” button, the PRA_Agent analyzes the given production request (depicted in Fig. 16) according to Section 4.1.

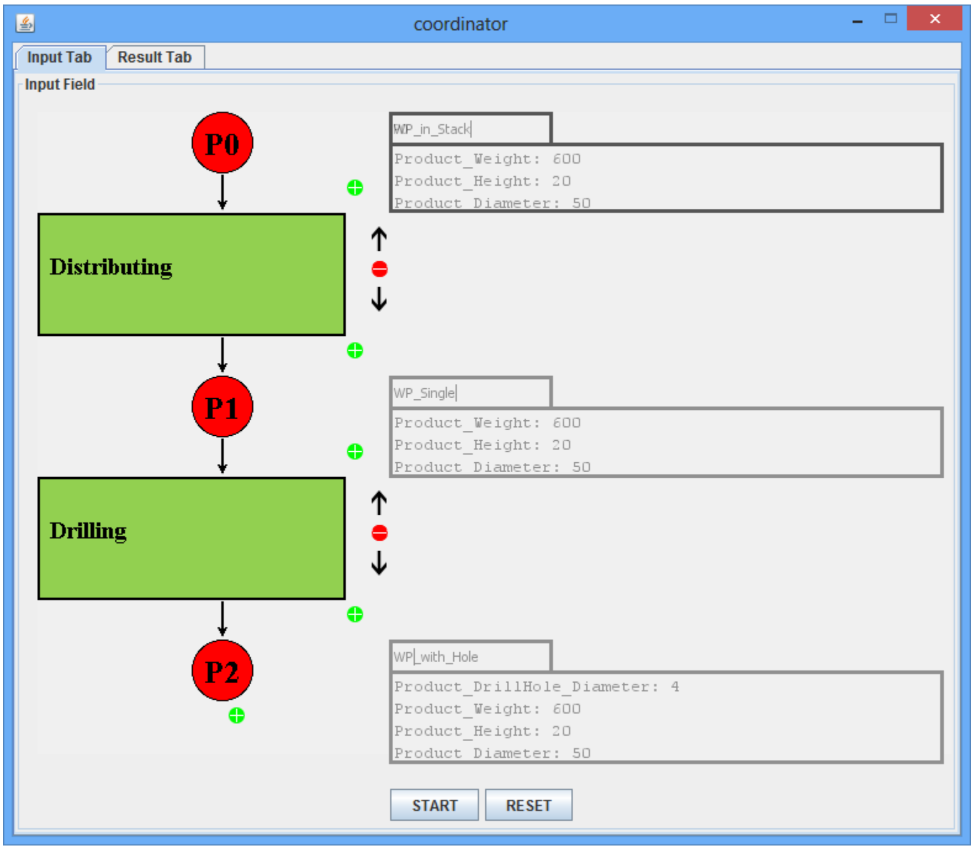


Fig 16. Production request for the multi agent system

By interaction with the MT_Agent, the PRA_Agent computes that the requested sequence of processes is feasible. By interaction with the FE_Agents of the four modules, it is determined that the parameter ranges of the process parameters and the corresponding resources do not fulfill the production request since the parameters Product_Weight and Product_Diameter are out of the current range of the related process parameters. The increase in weight affects the functional entities Distribution and Sorting/Storing which both use a vacuum gripper that does not offer enough suction force to transport the heavier workpieces. The increase in diameter affects all four stack magazines, material slides, the conveyor belt and the rotary indexing table.

In the following second phase of the adaptation process, adaptation options (represented by AO_Agents) to overcome the identified incapabilities are generated according to section 4.2. By analysis of the MDM of each FE_Agent, the AO_Agents analyze the change propagation of their adaptation option. The rooted tree graphs are constructed and displayed in the result tab of the GUI (see Fig. 17). Interaction with the SW_Agent reveals that a change of these parameters cannot be solely achieved by a change of the software.

For the sake of clarity, only possible adaptation options to overcome the limitations of the product weight in the Distributing station are discussed while explaining the generated rooted tree graphs. However, the GUI displays one rooted tree graph for each affected process parameter. To increase the process parameter WP-Weight_max, two independent resource parameters need to be adjusted. 1) For pushing the workpiece out of the stack magazine, the resource parameter ThrustCylinder_PushingForce needs to be increased. 2) For adjusting the vacuum gripper, the resource parameter VacuumCup_Force needs to be increased which can either be achieved by increasing the pressure of the vacuum cup or by increasing the diameter of the suction cup. The latter option has a negative influence on the process parameter WP-Diameter_min which might have an effect on the minimum product diameter.

For the stack magazines and the conveyor belts that were identified as too small (component-based restriction), a replacement of the component is suggested by the agent system. The parameter range of these elements cannot be adjusted since these restrictions are purely mechanical.

The adaptation options are analyzed by the EffortAspectAgent and FlexibilityAspectAgent in the following third phase of the adaptation process. The effort for replacement of the mechanical components can be determined by the approach described in section 4.3.3. The benefit can be measured in the suggested form of the bandwidth of process parameters. In case of the adaptations that allow a larger Product_Diameter, the upper level of the process parameter WP-Diameter_max is increased. For the conveyor belt, the minimum diameter of the work piece remains unchanged whereas in all resources where the exact position of the workpiece is determined by the mechanical component (e.g. the stack magazines), also WP-Diameter_min is increased by the replacement. In the latter case, the flexibility bandwidth of WP-Diameter is not increased since both boundaries are increased.

In the given example, the number of possible adaptation options is small enough to be overlooked by the user. None of the required parameter changes generated more than three adaptation options so that the complete tree graph was presented to the user in the GUI displayed in Fig. 17.

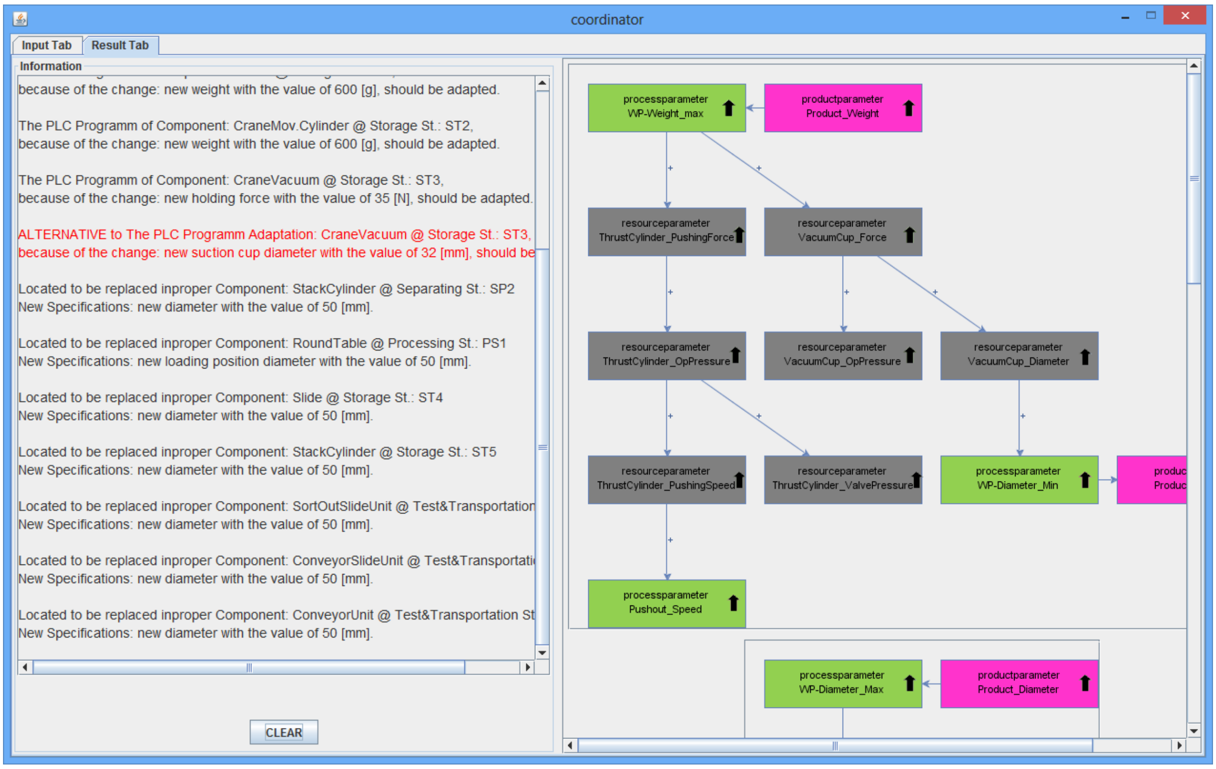


Fig. 17 Result tab for the production request with generated tree graphs

7 Summary and Outlook

The adaptation of existing automated manufacturing machines is a challenging task that occurs during the lifecycle of a machine. Currently, the adaptation process is performed manually and individually which is time-consuming, error-prone and is depending on the knowledge of the operating staff.

In this article, a systematized approach for the adaptation of automated manufacturing machines has been presented. The approach addresses the entire adaptation process, presenting different methods for each process phase and the corresponding challenges that occur when tool support shall be introduced in the adaptation process. Furthermore, an implementation concept using multi agent systems for a decision support system has been presented. The multi agent system implements the proposed approaches and algorithms and is able to propose suitable adaptation options based on a proposed PPR (product, process, resource) model of the machine that contains parameter ranges and interdependencies between system parameters and a given production request. The assistance and implementation concept has been applied to a lab-size modular production system. It has been shown that the adaptation process can be supported by the proposed concepts and that an automatic generation of adaptation options is possible.

By using the approaches and the decision support system presented in this article, a step towards a systematized and computer-supported adaptation process is performed. This has the future potential to

- decrease the planning time of adaptations since possible adaptation options are automatically generated using a model of the system
- increase the number of the considered solutions and thereby potentially increase the quality of the selected solution

- assist the operating staff in the adaptation process by processing knowledge and to make the adaptation process more independent from the knowledge of the operating staff

Future work will focus on the creation of the model of interdependencies between manufacturing machine elements since the results of the multi agent system heavily rely on the modeled information. This can be achieved by guidelines how to model the system in a suitable level of detail or by using tool-support for the (semi-)automatic determination of interdependencies. Furthermore, the article put its focus on the adaptation and flexibilization of process parameter ranges. However, a further flexibilization of a machine can also be achieved by adding new manufacturing skills and by adapting the material flow. In many cases, both options might need to go hand in hand. Further research in this field shall be conducted since the methodology for the generation of such adaptation options differs from the approach described in this article. Another aspect that will be integrated in future work is the usage of solution patterns that are retrieved from a repository. This repository can serve as an additional basis for the generation of adaptation options based on previously implemented adaptation options.

References

Ahmad N, Wynn DC, Clarkson PJ (2013) Change impact on a product and its redesign process: a tool for knowledge capture and reuse. Res Eng Design 24(3):219–244. doi: 10.1007/s00163-012-0139-8

Allen JD, Mattson CA, Thacker KS, Ferguson SM (2017) Design for excess capability to handle uncertain product requirements in a developing world setting. Res Eng Design 28(4):511–527. doi: 10.1007/s00163-017-0253-8

Beyer T, Göhner P, Yousefifar R, Wehking K (2016) Agent-based dimensioning to support the planning of Intra-Logistics systems. In: IEEE 21st International Conference on Emerging Technologies and Factory Automation. doi: 10.1109/ETFA.2016.7733647

Birkhofer R, Wollschlaeger M, Schrieper R, Winzenick M, Kalhoff J, Kleedörfer C, Mühlhause M, Niemann J (2012) Life-cycle-management for automation products and systems. A guideline by the system aspects working group of the ZVEI Automation Division. ZVEI, Automation Division, Frankfurt, M

Boehm BW (1981) Software engineering economics. Prentice-Hall, Englewood Cliffs, London

Boehm B, Clark B, Horowitz E, Westland C, Madachy R, Selby R (1995) Cost models for future software life cycle processes: COCOMO 2.0. Ann Software Eng 1(1):57–94. doi: 10.1007/BF02249046

Brennan RW, Norrie DH (2003) From FMs to HMs. In: Agent-Based Manufacturing. Springer, pp 31–49

Brill PH, Mandelbaum M (1989) On measures of flexibility in manufacturing systems. Int. J. of Prodn. Res. 27(5):747–756. doi: 10.1080/00207548908942584

Bürdek J, Kehrer T, Lochau M, Reuling D, Kelter U, Schürr A (2016) Reasoning about product-line evolution using complex feature model differences. Autom Softw Eng 23(4):687–733. doi: 10.1007/s10515-015-0185-3

Bussmann S, Jennings N, Wooldridge MJ (2010) Multiagent systems for manufacturing control. A design methodology. Springer series on agent technology. Springer, Berlin, London

Buzacott J. A., Mandelbaum M. (2008) Flexibility in manufacturing and services: achievements, insights and challenges. Flex Serv Manuf J 20(2):13–58. doi: 10.1007/s10696-008-9052-x

Cavin S, Lohse N (2004) Multi-level skill-based allocation methodology for evolvable assembly systems. In: 12th IEEE International Conference on Industrial Informatics (INDIN). doi: 10.1109/INDIN.2014.6945569

Clarkson PJ, Simons C, Eckert C (2004) Predicting Change Propagation in Complex Design. J. Mech. Des. 126(5):788. doi: 10.1115/1.1765117

Das SK (1996) The measurement of flexibility in manufacturing systems. Int J Flex Manuf Syst 8(1):67–93. doi: 10.1007/BF00167801

Das SK, Yedlarajiah P, Narendra R (2000) An approach for estimating the end-of-life product disassembly effort and cost. International Journal of Production Research 38(3):657–673. doi: 10.1080/002075400189356

ElMaraghy HA (2005) Flexible and reconfigurable manufacturing systems paradigms. Int J Flex Manuf Syst 17(4):261–276. doi: 10.1007/s10696-006-9028-7

ElMaraghy HA, AlGeddawy T (2012) Co-evolution of products and manufacturing capabilities and application in auto-parts assembly. Flex Serv Manuf J 24(2):142–170. doi: 10.1007/s10696-011-9088-1

Engel A, Browning TR (2008) Designing systems for adaptability by means of architecture options. Syst. Engin. 11(2):125–146. doi: 10.1002/sys.20090

Engel A, Browning TR, Reich Y (2017) Designing Products for Adaptability: Insights from Four Industrial Cases. Decision Sciences 48(5):875–917. doi: 10.1111/deci.12254

Engel A, Reich Y (2015) Advancing Architecture Options Theory: Six Industrial Case Studies. Syst. Engin. 18(4):396–414. doi: 10.1002/sys.21312

Enparantza R, Revilla O, Azkarate A, Zendoia J (2006) A life cycle cost calculation and management system for machine tools. Proceedings of LCE2006:717–722

Fei G, Gao J, Owodunni O, Tang X (2011) A method for engineering design change analysis using system modelling and knowledge management techniques. International Journal of Computer Integrated Manufacturing 24(6):535–551. doi: 10.1080/0951192X.2011.562544

Frayret J, D'Amours S, Rousseau A, Harvey S, Gaudreault J (2007) Agent-based supply-chain planning in the forest products industry. Int J Flex Manuf Syst 19(4):358–391. doi: 10.1007/s10696-008-9034-z

Gharieb W (2006) Software Quality in Ladder Programming. In: International Conference on Computer Engineering and Systems. doi: 10.1109/ICCES.2006.320440

Halstead MH (1977) Elements of software science. Elsevier computer science library, vol 2. Elsevier, New York, Oxford [etc.]

Hamraz B, Caldwell NHM, Wynn DC, Clarkson PJ (2013) Requirements-based development of an improved engineering change management method. Journal of Engineering Design 24(11): 765-793. doi: 10.1080/09544828.2013.834039

Harivardhini S, Chakrabarti A (2016) A new model for estimating End-of-Life disassembly effort during early stages of product design. Clean Techn Environ Policy 18(5):1585–1598. doi: 10.1007/s10098-016-1142-y

Haubeck C, Wior I, Braubach L, Pokahr A, Ladiges J, Fay A, Lamersdorf W (2013) Keeping pace with changes-towards supporting continuous improvements and extensive updates in production automation software. Electronic Communications of the EASST 56

Hoang XL, Fay A, Marks P, Weyrich M (2016) Systematization approach for the adaptation of manufacturing machines. In: IEEE ETFA 2016. doi: 10.1109/ETFA.2016.7733635

Hoang XL, Marks P, Fay A, Weyrich M (2017, in press) Generation and Impact Analysis of Adaptation Options for Automated Manufacturing Machines. In: IEEE ETFA 2017

Hoang X, Marks P, Weyrich M, Fay A (2017) Modeling of interdependencies between products, processes and resources to support the evolution of mechatronic systems. IFAC-PapersOnLine 50(1):4348–4353. doi: 10.1016/j.ifacol.2017.08.873

Hu J, Cardin M (2015) Generating flexibility in the design of engineering systems to enable better sustainability and lifecycle performance. Res Eng Design 26(2):121–143. doi: 10.1007/s00163-015-0189-9

Idri A, Amazal Fa, Abran A (2015) Analogy-based software development effort estimation: A systematic mapping and review. Information and Software Technology 58:206–230. doi: 10.1016/j.infsof.2014.07.013

Järvenpää E (2012) Capability-based adaptation of production systems in a changing environment. Dissertation, University of Technology, Tampere

Jarvenpää E, Siltala N, Lanz M (2016) Formal resource and capability descriptions supporting rapid reconfiguration of assembly systems. In: IEEE International Symposium on Assembly and Manufacturing (ISAM). doi: 10.1109/ISAM.2016.7750724

Karl F, Reinhart G (2015) Reconfigurations on manufacturing resources: identification of needs and planning. Prod. Eng. Res. Devel. 9(3):393–404. doi: 10.1007/s11740-015-0607-x

Kasarda ME, Terpenny JP, Inman D, Precoda KR, Jelesko J, Sahin A, Park J (2007) Design for adaptability (DFAD)—a new concept for achieving sustainable design. Robotics and Computer-Integrated Manufacturing 23(6):727–734. doi: 10.1016/j.rcim.2007.02.004

Keller R, Eger T, Eckert C, Clarkson PJ (2005) Visualising change propagation. In: 15th Int. Conf on Eng Design

Koch J, Gritsch A, Reinhart G (2016) Process design for the management of changes in manufacturing: Toward a Manufacturing Change Management process. CIRP Journal of Manufacturing Science and Technology 14:10–19. doi: 10.1016/j.cirpj.2016.04.010

Kochikar VP, Narendran TT (1992) A framework for assessing the flexibility of manufacturing systems. International Journal of Production Research 30(12):2873–2895. doi: 10.1080/00207549208948196

Ladiges J, Wior I, Arroyo E, Fay A, Haubeck C, Lamersdorf W (2013a) Evolution of production facilities and its impact on non-functional requirements. In: IEEE 11th International Conference on Industrial Informatics (INDIN). doi: 10.1109/INDIN.2013.6622886

Ladiges J, Fay A, Haubeck C, Lamersdorf W (2013b) Operationalized definitions of non-functional requirements on automated production facilities to measure evolution effects with an automation system. In: IEEE 18th Conference on Emerging Technologies & Factory Automation. doi: 10.1109/ETFA.2013.6648092

Ladiges J, Fay A, Lamersdorf W (2016) Automated Determining of Manufacturing Properties and Their Evolutionary Changes from Event Traces. Intell Ind Syst 2(2):163–178. doi: 10.1007/s40903-016-0048-7

Leitao P (2015) Industrial Agents. Elsevier Science

Leitao P, Barbosa J, Vrba P, Skobelev P, Tsarev A, Kazanskaia D (2013) Multi-agent System Approach for the Strategic Planning in Ramp-Up Production of Small Lots. In: IEEE International Conference on Systems, Man and Cybernetics (SMC 2013). doi: 10.1109/SMC.2013.807

Lucas M, Tilbury D (2005) Methods of measuring the size and complexity of PLC programs in different logic control design methodologies. Int J Adv Manuf Technol 26(5-6):436–447. doi: 10.1007/s00170-003-1996-0

Maga CR, Jazdi N, Göhner P (2011) Reusable Models in Industrial Automation: Experiences in Defining Appropriate Levels of Granularity. IFAC Proceedings Volumes 44(1):9145–9150. doi: 10.3182/20110828-6-IT-1002.01509

Marks P, Hoang XL, Fay A, Weyrich M (2017, in press) Agent-based Adaptation of automated Manufacturing Machines. In: IEEE ETFA2017

Marks P, Weyrich M (2017) Assistenzsystem zur Aufwandsabschätzung der Software-Evolution von automatisierten Produktionssystemen. In: Automation 2017 - Technology networks Processes

McCabe T (1976) A Complexity Measure. IIEEE Trans. Software Eng. 2(4):308–320. doi: 10.1109/TSE.1976.233837

Ollinger GA, Stahovich TF (2004) RedesignIT—A Model-Based Tool for Managing Design Changes. J. Mech. Des. 126(2):208. doi: 10.1115/1.1666888

Ouertani MZ, Yesilbas LG, Lossent L (2004) Engineering change process: state of the art, a case study and proposition of an impact analysis method. In: 5th Int. Conf. on Integrated Design and Manufacturing in Mechanical Engineering

Beitz W, Pahl G, Beitz W (1997) Konstruktionslehre: Methoden und Anwendung. Springer Berlin

Pfrommer J, Stogl D, Aleksandrov K, Escaida Navarro S, Hein B, Beyerer J (2015) Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. at - Automatisierungstechnik 63(10). doi: 10.1515/auto-2014-1157

Reddi KR, Moon YB (2009) A framework for managing engineering change propagation. International Journal of Innovation and Learning 6(5):461–476

Rogalski S (2011) Flexibility Measurement in Production Systems. Handling Uncertainties in Industrial Production. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg

Schleipen M, Drath R (2009) Three-view-concept for modeling process or manufacturing plants with AutomationML. In: IEEE ETFA 2009. doi: 10.1109/ETFA.2009.5347260

Seidl C, Teixeira L, Lity S, Kowal M, Schaefer I (2017) Higher-order delta modeling for software product line evolution. In: 7th Int. Workshop on Feature-Oriented Software Development

Sethi A, Sethi S (1990) Flexibility in manufacturing: A survey. Int J Flex Manuf Syst 2(4). doi: 10.1007/BF00186471

Tompkins JA (2010) Facilities planning, Fourth edition. John Wiley & Sons, Hoboken, N.J

Ulewicz S, Ulbrich M, Weigl A, Kirsten M, Wiebe F, Beckert B, Vogel-Heuser B (2016) A verification-supported evolution approach to assist software application engineers in industrial factory automation. In: IEEE ISAM, pp 19–25

VDI (2017) Adaptability: Description and measurement of the adaptability of manufacturing companies (Medical device industry) (VDI 5201 Part 1)

VDI/VDE Society for Measurement and Automatic Control (2015) Formalised process descriptions (VDI/VDE 3682)

Vogel-Heuser B, Fay A, Schaefer I, Tichy M (2015) Evolution of software in automated production systems: Challenges and research directions. Journal of Systems and Software 110:54–84. doi: 10.1016/j.jss.2015.08.026

Wahab M, Wu D, Lee C (2008) A generic approach to measuring the machine flexibility of manufacturing systems. European Journal of Operational Research 186(1):137–149. doi: 10.1016/j.ejor.2007.01.052

Wenzel S, Poggenpohl D, Jürjens J, Ochoa M (2014) Specifying model changes with UMLchange to support security verification of potential evolution. Computer Standards & Interfaces 36(4):776–791. doi: 10.1016/j.csi.2013.12.011

Woods DD (1986) Paradigms for Intelligent Decision Support. In: Hollnagel E, Mancini G, Woods DD (eds) Intelligent Decision Support in Process Environments, vol 21. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 153–173

Wooldridge M, Jennings NR (1995) Intelligent agents: Theory and practice. The knowledge engineering review 10(2):115–152

Wooldridge M, Jennings NR, Kinny D (2000) The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems 3(3):285–312. doi: 10.1023/A:1010071910869

Yang F, Duan G (2012) Developing a parameter linkage-based method for searching change propagation paths. Res Eng Design 23(4):353–372. doi: 10.1007/s00163-011-0124-7

Younis MB, Frey G (2007) Software quality measures to determine the diagnosability of PLC applications. In: IEEE ETFA. doi: 10.1109/EFTA.2007.4416791

Zäh MF, Reinhart G, Lindemann U, Karl F, Biedermann W (2011) DSM-based Evaluation of Assembly Manufacturing Ressources. In: 13th Int DSM Conference, pp 435–448

Zhang J, Xue G, Du H, Garg A, Peng Q, Gu P (2017) Enhancing interface adaptability of open architecture products. Res Eng Design 28(4):545–560. doi: 10.1007/s00163-017-0264-5