20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Context-oriented Modularization of Product Development Processes using Matrix-Based Clustering

Christoph Hollauer[1], Robin Thomas[1], Donna H. Rhodes[2], Udo Lindemann[1]

[1]Laboratory for Product Development and Lightweight Design, TUM
[2]Systems Engineering Advancement Research Initiative, MIT

Abstract: System modularization is a common and well-established approach to reduce system complexity. However, methodical approaches for the modularization of product development processes (PDPs) can hardly be found in the literature. The work that exists focuses only on interdependencies between process elements when modularizing the process. This paper proposes a modularization method for PDPs, basing the modularization on the context of the respective process, while still also taking process-internal interdependencies into account. The matrix-based approach applies a clustering algorithm that uses process context data to group process elements into modules. The modular PDP can then be tailored into project-specific PDPs based on the project context at hand. The design and application of lean and efficient project-specific PDPs has promising potential to reduce product development effort and costs.

Keywords: Product development process, modularization, clustering, context, modular processes, tailoring

1 Introduction

Technology companies around the world face challenges like rapidly increasing product and service complexity, increasing customer requirements and numbers of stakeholders involved, as well as shorter development- and product life-cycles (Allweyer 2005, Browning and Ramasesh 2007, Junge 2013, Fischer 2015). Therefore, product development processes (PDPs) that provide the desired outcomes for different product development situations and project scopes in a quick and efficient manner gain more and more importance and are a key aspect of success for every company involved in product development (Sered and Reich 2006, Cooper 2014). As PDPs are highly influenced by the boundary conditions of the development situation at hand and the corresponding specific requirements (Roelofsen 2009), it is recommended to always design the process with its application context in mind. Rosemann and Recker defined a company´s context as the combination of all situational circumstances that impact process design and execution (Rosemann and Recker 2006). Considering a company´s context, a modular PDP can be designed (Rosemann and Recker 2006), which can then be tailored into efficient project-specific PDPs by applying guidelines based on the project-specific situation with all its requirements and constraints (Ginsberg and Quinn 1995, Hollauer and Lindemann 2017). Ginsberg and Quinn define tailoring as “the act of adjusting the definition and/or particularizing the terms of a general description to an alternate environment”, which for the area of product development can be interpreted as the adaption of a company’s standard set of processes to specific project contexts defined by particular context variables (Ginsberg and Quinn 1995, Hollauer and Lindemann 2017).

This paper presents a methodical approach for modularizing an existing PDP based on the project contexts it is to be tailored to, thereby deducing a modular process that is tailorable into project-specific processes. The following section defines the objectives of the development of the modularization approach and presents necessary theoretic groundwork based on an in-depth literature review. Subsequently, the modularization approach itself and its first evaluation via case studies conducted with an implemented software prototype are detailed. In a final step, potential future research regarding the modularization approach is outlined.

2 Background and Objectives

This section provides a brief overview of modularity in general and modularization in the areas of processes and products to then derive the objectives of the context-oriented modularization approach for PDPs.

2.1 Modularity

A broad variety of definitions of modularity can be found in the literature. Reijers, Mendling et al. (2010) propose a very general definition, stating that modularity is commonly interpreted as the design principle of having a complex system composed from smaller subsystems, that can be managed independently yet function together as a whole. Göpfert (1998) and Bauer (2016) describe modularity as an approach to reduce the complexity of a system by dividing it into smaller subsystems or modules, that minimize interfaces between each other, but have a high degree of interaction within each module. The concept was first used for product modularization in order to be able to understand and control the steadily growing complexity of products, preceding its application on processes (Göpfert 1998, Langlois 2002). Besides the reduction of system complexity as the general motivation for modularity, further advantages are: standardization, decoupling, combinability, flexibility, reuse, efficiency, controllability, replaceability, changeability and adaptability (Sanchez and Mahoney 1996, Gu et al. 1997, Göpfert 1998, Gu and Sosale 1999, Renner 2007, Seol et al. 2007, Krause and Ripperda 2013).

2.2 Existing approaches

A literature review regarding basic information and existing approaches for process modularization was conducted with regard to PDPs (focus area) and business processes (BP). BP are usually less complex, less parallel, include less iterations and have less complex interdependencies within the process (Browning et al. 2006, Lindemann 2009, Clarkson and Eckert 2010, Koch 2015), making modularization easier and more common.

PDP modularization: The investigation of modularity in the area of PDPs identified a number of methods and approaches for flexible design of PDPs due to the respective development situation. Examples are a method of modelling PDPs using process blocks (Bichlmaier et al. 1999), relation-oriented process synthesis (Baumberger 2007), the FORFLOW process model (Roelofsen 2011), and the Stage-Gate approach (Cooper

2001). All of these and other existing methods in the literature are well-established tools providing useful general information and guidelines for the design of flexible PDPs based on different development situations. However, no methodology for modularizing an existing process based on a company's different project contexts could be found. The only method coming close is the concept for design process modularization proposed by Seol et al. (2007). The authors divide an existing PDP into its constituent activities and cluster (group) these into modules via an algorithm analyzing the process flow between the activities in a design structure matrix (DSM) (Seol et al. 2007). The modularization approach presented in this paper similarly divides the overall process into activities and groups them into modules with a clustering algorithm, but the method proposed by Seol et al. (2007) could not be used as the basis for the development of a context-oriented modularization method. The reason for this is that the algorithm they apply is too limited and the modularization is purely based on the process flow, not taking project contexts into account. Nevertheless, ideas, requirements and restrictions could be derived from that concept. As there is no methodical approach for a context-oriented modularization of existing PDPs, a second step was to analyze modularity in the area of business processes, where its application is more established.

Business process modularization: The main purposes of the modularization of business process are to increase process understanding among the stakeholders (Gruhn and Laue 2006, Mendling et al. 2010), to support communication (Reijers and Mendling 2008, Melissen 2013), and to take advantage of reuse of already existing modules (Gruhn and Laue 2006, Reijers and Mendling 2008). However, in general, the focus of research on business process modularization is of conceptual nature and there are no objective and explicit guidelines, tool support or methodical approaches, that modelers in practice can rely on (Reijers and Mendling 2008, Mendling et al. 2010). The idea of basing process design on the context of a company is outlined by Rosemann and Recker (2006), who suggest designing flexible, context-oriented business processes, but do not propose any kind of methodical approach. To summarize, modularization approaches for business processes that could be applied on PDPs considering the company context are also currently lacking.

Product modularization: With limited existing modularization methods for PDP or business processes to base on, the decision was made to investigate methods for product modularization for adaptation to the use on PDPs. This was chosen as a significant number of elaborated modularization methods for products are readily available, and the application of product modularization is a very common approach (Krause and Ripperda 2013). After an in-depth review regarding existing product modularization methods and a detailed comparison of the eight most promising approaches, the extended modular function deployment proposed by Stake (Stake 2000) was found most promising to be adapted and extended into a context-oriented modularization approach for PDPs.

2.3 Research gap and objectives

Frameworks and guidelines for the design of flexible PDPs already exist, but methodical support for the modularization of an existing PDP based on the process context is currently limited. A modular process could subsequently be tailored into project-specific processes more easily. This paper aims to contribute to closing this research gap by elaborating a corresponding modularization approach, focusing on the following objectives:

- The actual modularization of the PDP should be completely based on the respective company's context data as the key novel aspect of the method.
- The method should take interfaces between process activities into account, to allow a comparison of the quality of different modularization scenarios.
- The method should not be limited to a specific industry sector and specific type or complexity of PDP, in order to maximize the applicability of the method.
- The method should be implemented in a software demonstrator as a proof of concept.
- The software demonstrator should be applied and evaluated using case studies.

3 Proposed method

Figure 1 displays the steps of the final context-oriented modularization method for PDPs and its modularization algorithm after several steps of adapting, extending and modifying the modular extended modular function deployment that served as the basic framework. The individual steps are subsequently detailed.

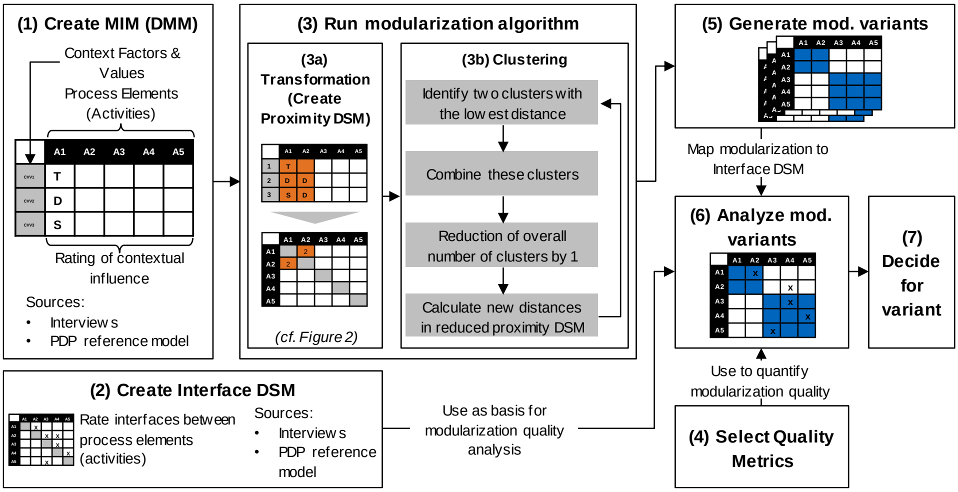


Figure 1: Steps of the modularization method and the modularization algorithm

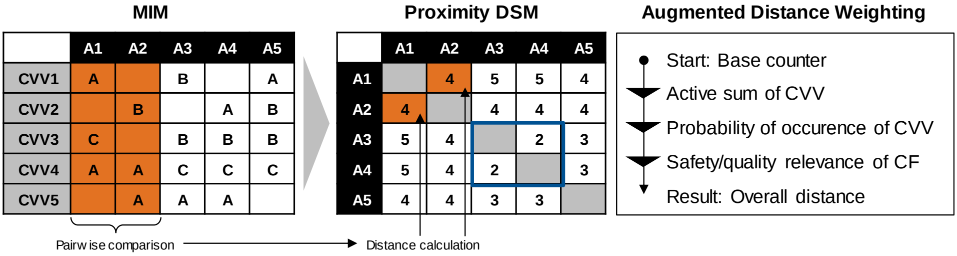
(1) Create MIM (Rate activities regarding contextual influence): The modularization is based on the company's project contexts, which can be defined as all internal and external boundary conditions influencing the development activities within the respective company. The context is documented in the form of context variables with different values to describe the possible project contexts of a specific company. Examples include the different types of projects that are conducted within a company, the disciplines involved, the industries and markets it is doing business in, the complexity of its product portfolio, etc. The PDP is documented in a reference process model containing, among other, the process activities Subsequently, all process elements (activities) are rated regarding the influence of each context variable value on them in the so-called module indication matrix (MIM). The MIM is a domain mapping matrix (DMM), containing the process elements (columns) from the reference PDP model and the context variable values (rows), acquired using e.g. interviews. The rating regarding the influence of each context value on each process element can be performed via a numerical rating system (strength) or using qualitative operators that are later applied for the tailoring of project-specific processes. For example, for a particular context value an activity “must be tailored”, “is deleted”, or “a specific mode selected” (cf. “T”, “D”, “S” in Figure 1). The rated MIM forms the basis for the modularization algorithm, by comparing the similarity of the ratings of contextual influence on process elements, grouping elements into modules that have similar context ratings and will therefore be necessary in the same project context.

(2) Create Interface DSM (Rate activity interfaces): To consider dependencies between process elements (activities), their interfaces are documented in a design structure matrix (DSM). Different types of process interfaces can be considered, based on an interface catalog derived from literature, e.g. interfaces regarding collaboration, communication, information, and organization. The interfaces must be defined and rated by organizational process experts. The only requirement regarding the rating system applied to quantify the intensity of process interfaces is that it has to be numerical. The completely rated interface DSM (iDSM) forms the decision basis to assess the quality of the various modularization variants generated by the modularization algorithm. As in-depth process knowledge is required to perform both, rating in the MIM and the iDSM, company-internal process experts should rate their respective process activities regarding context influences and interfaces for the application of the method.

(3) Run modularization algorithm: With the rated MIM, a two-step modularization algorithm is run to generate possible modularization variants. In a first step, the MIM is transformed into a symmetrical proximity matrix (pDSM), with which the actual modularization is performed (see figure 2 for a simplified example). The clustering algorithm applied in the software prototype and case studies is a *hierarchical, agglomerative* clustering algorithm, which was selected and designed based on the guidelines for the elaboration of clustering algorithms in Backhaus et al. (2015). The application of other clustering algorithms is possible as well, as long as they use the similarity of the process element´s context ratings as the clustering criteria.

(3a) Transformation algorithm: To run the clustering algorithm, the MIM is first transformed into a pDSM (process elements x process elements), containing the distance of the ratings of the process elements from the MIM. During the transformation, each activity pair is compared and the calculated distance documented in the respective cell. For each pair, non-identical context variable values increase the distance by the value “1” (cf. Figure 2, orange highlights). This basic counter can be augmented through multiplication and addition of the basic counter with a pre-defined weighting system (Figure 2, right). The weighting system can be adapted to the situation at hand to increase or decrease the influence of context variable values on the modularization. Possible elements of the weighting element are: the active sum of context variable values in order to increase the weight of influential values, the probability of occurrence for individual context values, or modified distance counters for safety/quality relevant context factors.

(3b) Clustering algorithm: The selected clustering algorithm is subsequently applied on the resulting pDSM (cf. Figure 1). The algorithm starts with the assumption that each matrix element (process activity) forms its own module. In every step, the two elements/clusters with the smallest distance regarding the influence rating of the different context variable values are grouped together into a cluster and the overall number of clusters is reduced by one. Afterwards, the distances of the newly formed cluster to all other existing clusters are updated, leading to a reduced pDSM, upon which the next algorithm step will be executed on. For this step either the smaller (single linkage, SL) or the higher distance (complete linkage, CL) of the two distances of the clusters being combined can be assessed as the new distance to each other element/cluster, leading to different possible modularizations. Each of the procedures or a combination of both can be favorable under certain circumstances, but further research regarding this aspect is necessary. The steps are repeated until a previously defined number of clusters (i.e. modules) is reached. This way, several possible modularizations can be generated and compared to identify the solution with the highest modularization quality due to quality metrics.



MIM

	A1	A2	A3	A4	A5
CVV1	A		B		A
CVV2		B		A	B
CVV3	C		B	B	B
CVV4	A	A	C	C	C
CVV5		A	A	A	

Proximity DSM

	A1	A2	A3	A4	A5
A1		4	5	5	4
A2	4		4	4	4
A3	5	4		2	3
A4	5	4	2		3
A5	4	4	3	3	

Augmented Distance Weighting

Start: Base counter
Active sum of CVV
Probability of occurence of CVV
Safety/quality relevance of CF
Result: Overall distance

Figure 2: Simplified example of the transformation and clustering algorithm

(4) Iterate to generate modularization variants: In order to identify a high-quality modularization, several iterations of the modularization algorithm are necessary to generate variants for comparison. The preferred modularization solution, showing the highest modularization quality, is subsequently selected using the applied modularization quality metrics.

(5) Select modularization quality metrics: Structural metrics are employed for the comparison of different potential modularizations generated by the algorithm. The metrics are applied on the modularized iDSM. Most modularization quality metrics determine the modularization quality based on an analysis of the interfaces between and within the modules. Additionally, metrics focusing on different modularity aspects, e.g. the number of involved stakeholders per modules, are feasible as well. The metrics should be chosen based on the respective situation and the desired focus. Examples of possible metrics supporting the analysis of the modularization in the iDSM are:

- *Cluster perspective/module density* (Behncke 2017, Koppenhagen 2004): Minimizing unwanted interfaces between modules that can limit the success of modularization.
- *System perspective/module independence index* (Behncke 2017, Koppenhagen 2004): Maximizing necessary interfaces within modules.

- *Module qualities* (Kreimeyer 2009): Analyzing the compactness of modules (interaction of a module with its environment) and the flow of information between them, both of which should be limited for a high-quality modularization.
- *Stakeholder metrics*: Limiting the number of involved stakeholders per module or the number of modules one particular stakeholder is involved in.

One aspect the quality metrics should always consider is the overall heterogeneity of the modules that increases with a decreasing number of modules and increasing number of elements per module.

(6) Analyze modularization variants by calculating quality metrics: The metrics chosen for the analysis of the potential modularizations in a specific context are calculated and compared to provide the data basis for deciding on one of the modularizations. Before applying the quality metrics, each possible modularization generated by the clustering algorithm must be transferred to the iDSM.

(7) Decide on one modularization: Based on the results of the variant analysis using the quality metrics, the last step is making a decision for the design of the modular PDP with the highest quality due to the quality metrics.

4 Evaluation

The modularization method is implemented in a Microsoft Excel-based software prototype programmed using visual basic for applications (VBA). The software prototype was subsequently applied on two case studies to verify the overall approach, including the modularization algorithm as well as the quality metrics. This was done to ensure the algorithm is functioning as intended and provides valid results that comply with the objective of deducting a context-specific modularization of an existing PDP.

The first case study was conducted with a small, academic set of input data with low complexity. In both case studies, modularization variations were automatically generated by the modularization algorithm implemented in the software prototype and manually compared by the authors applying the modularization quality metrics. Figure 3 shows an example of a modularization during the first case study displayed in the MIM. For the academic case study, no expert-based independent evaluation of the results was possible.

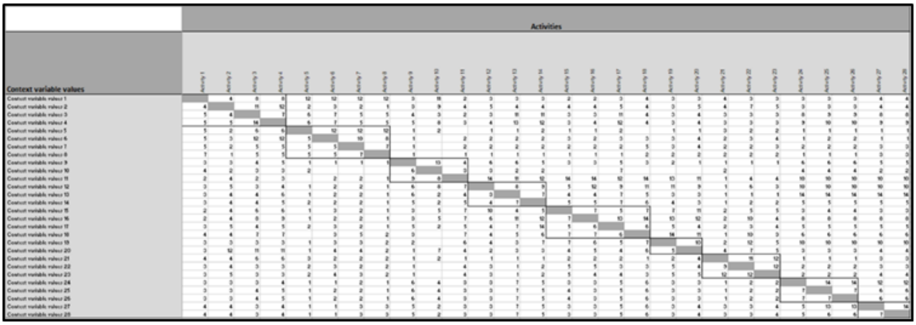


Figure 3: Modularization example for the academic case study

The second case study was based on data from an industrial case study (medium-sized plant engineering company), where 218 process activities and 231 context values with corresponding MIM ratings have been obtained, but only partial data regarding the activity interface ratings (iDSM) was available. The MIM data included ratings in a quantitative form, indicating whether an activity can be dropped, needs to be carried out, or needs to be carried out intensively, depending on the context values for a particular project. Due to confidentiality concerns, this data cannot be published. After performing test runs with both SL and CL algorithms, the heterogeneity curves of the resulting modularization were analyzed, but, due to their similarity, did not provide a conclusive lead for the selection of an algorithm. Also, no optimal number of clusters due to the "elbow-criterion" could be identified (cf. Backhaus et al 2015, pp. 494-496). The eventual clustering of the calculated pDSM was subsequently carried out in two stages to derive 20, 30, 40, and 50 clusters: First, a SL algorithm generated 10 clusters consisting of only one to three elements, with another cluster containing the remaining activities. Removing these cluster, the remaining larger cluster was "sub-modularized" using a CL algorithm. The subsequent metric analysis indicated that the combination of SL and CL algorithms with a cluster count of 20 produced the modularization of the highest quality (not regarding the homogeneity of clusters). However, the choice of algorithm strongly depends on the intended number of clusters, as the two-stage approach only produced the best results for 20 clusters. For higher numbers of clusters, the differences between the combined approach and a single stage CL algorithm were negligible. In fact, if the objective is to derive more homogeneous clusters, for 50 clusters the CL algorithm produced slightly better results, and also requires less effort. The number of intended modules should be defined with the overall number of process activities in mind, setting the number of modules to e. g. 10 to 25% of the overall process activities.

For the second case study, a detailed evaluation with the process expert responsible for the elaboration of the context and process model was performed. The process expert confirmed the usefulness of the modularization metrics and the validity of the results. The most important aspect he pointed out, was the selection of the applied quality metrics. They must be selected carefully regarding the the key objectives of the modularization in a specific situation (e.g. avoiding upstream interfaces possibly causing rework, minimizing the number of stakeholders involved per module, minimizing the overall flow of information between modules, etc.) to assure finding the optimal solution. Therefore, internal process experts should select the quality metrics to apply in the decision-making process, as well as which interfaces to consider for the interface analysis in the iDSM.

To summarize, the case studies showed that the modularization approach provides the necessary tools and guidelines for a context-oriented modularization of an existing PDP and verified the usefulness of the quality metrics for supporting the decision for one of several possible modularizations generated by the modularization algorithms. Additionally, the case studies revealed promising areas of further elaboration of the modularization method. However, the approach is currently considered preliminary and requires further testing and refinement.

5 Conclusion and future research

In this paper we have presented a preliminary approach for an algorithmic, matrix-based modularization of PDPs based on differing project contexts. The approach considers relevant process interfaces by basing the modularization quality assessment on the interface analysis of the modularized PDP. The thus modularized PDP is expected to be more easily tailorable due to the grouping of similarly influenced activities. The process modules serve as a basis for grouping and managing activities subject to similar contextual influences. To summarize, the identified research gap can be addressed by the developed approach, as it reproducibly generates a modular PDP, that capitalizes on the advantages of modularity, such as. adaptability and flexibility. Subsequent tailoring of the reference PDP can avoid unnecessary activities and therefore reduce time, effort and cost. This tailoring step can, for example, be performed in collaborative workshops with project stakeholders. The presented approach can contribute to reducing the process tailoring effort, which is a crucial advantage in times of strong competitiveness in globalized markets and steadily increasing importance of efficiency (Sered and Reich 2006, Fischer 2015). The current state of the developed method represents a basis for further experimentation with a high potential for further elaboration and application in industry.

Additional case studies need to be conducted for further evaluation and refinement, with different input data and boundary conditions. The following aspects should be tested and compared in particular: Different rating systems for the assessment of the influence of the context variable values on process elements (activities) in the MIM, weighting systems for the transformation of the MIM into a DSM, different clustering algorithms, and the significance of the quality metrics for practitioners. The base approach itself is designed to be adaptable to such changes.

Another area for future research is the improvement of the software prototype, both in terms of performance as well as automation of the decision-making process by including the quality metrics in the algorithm to combine the generation of module variations and their analysis (closed-loop optimization). So far, this needs to be done manually, but with an enhanced software tool the user could define quality metrics and weighting system beforehand, with the software automatically generating the modularization solution space, identifying the best solution automatically. Another aspect not yet covered is how to keep the resulting modular PDP up to date and adapt it to significant changes in the context.

References

Allweyer, T. 2005. Geschäftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling, W3l GmbH.

Backhaus, K., et al. 2015. Multivariate analysemethoden: eine anwendungsorientierte einführung, Springer-Verlag.

Bauer, W. 2016. Planung und Entwicklung änderungsrobuster Plattformarchitekturen, Unpublished Dissertation, Verlag Dr. Hut.

Baumberger, G. C. 2007. Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten, Verlag Dr. Hut.

Behncke, F. G. H. 2017. Beschaffungsgerechte Produktentwicklung, Unpublished Dissertation.

Bichlmaier, C., et al. 1999. PMM-Process Module Methodology for Integrated Design and Assembly Planning. Proceedings of the 4th Design for Manufacturing Conference, Las Vegas.

Browning, T. R., et al. 2006. Key concepts in modeling product development processes. Systems Engineering 9(2): 104-128.

Browning, T. R. and R. V. Ramasesh 2007. A survey of activity network- based process models for managing product development projects. Production and operations management 16(2): 217-240.

Clarkson, J. and C. Eckert 2010. Design process improvement: a review of current practice, Springer Science & Business Media.

Cooper, R. G. 2001. Winning at New Products: Accelerating the Process from Idea to Launch.

Cooper, R. G. 2014. What's Next?: After Stage-Gate. Research-Technology Management **57**(1): 20-31.

Fischer, U., Müller, P. 2015. Mega-Trends und ihr Einfluss auf die Produktentwicklung.

Ginsberg, M. P. and L. H. Quinn 1995. Process tailoring and the the software capability maturity model.

Göpfert, J. 1998. Modulare Produktentwicklung. Innovationsforschung und Technologiemanagement, Springer: 139-151.

Gruhn, V. and R. Laue 2006. Complexity metrics for business process models. 9th international conference on business information systems (BIS 2006).

Gu, P., et al. 1997. An integrated modular design methodology for life-cycle engineering. CIRP Annals-Manufacturing Technology 46(1): 71-74.

Gu, P. and S. Sosale 1999. Product modularization for life cycle engineering. Robotics and Computer-Integrated Manufacturing 15(5): 387-401.

Hollauer, C. and U. Lindemann 2017. Design Process Tailoring: A Review and Perspective on the Literature. International Conference on Research into Design, Springer.

Junge, M. 2013. Modularisierung in der Automobilindustrie. Perspektiven und Facetten der Produktionswirtschaft: Schwerpunkte der Mainzer Forschung: 89.

Koch, S. 2015. Einführung in das Management von Geschäftsprozessen: Six Sigma, Kaizen und TQM, Springer-Verlag.

Koppenhagen, F. 2004. Systematische Ableitung modularer Produktarchitekturen. Berichte aus der Konstruktionstechnik, Shaker, Aachen: 16.

Krause, D. and S. Ripperda 2013. An assessment of methodical approaches to support the development of modular product families. 19th International Conference on Engineering Design (ICED13), Seoul.

Kreimeyer, M. F. 2009. A structural measurement system for engineering design processes. Dr. Hut, München.

Langlois, R. N. 2002. Modularity in technology and organization. Journal of economic behavior & organization 49(1): 19-37.

Lindemann, U. 2009. Methodische Entwicklung technischer Produkte. 3., korrigierte Auflage, Berlin, Heidelberg (Springer).

Melissen, R. R. 2013. A business process harmonization approach: using process mining techniques.

Mendling, H. R. J. and R. Dijkman 2010. On the Usefulness of Subprocesses in Business Process Models.

Reijers, H. and J. Mendling 2008. Modularity in Process Models: Review and Effects. Business Process Management: 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings. M. Dumas, M. Reichert and M.-C. Shan. Berlin, Heidelberg, Springer Berlin Heidelberg: 20-35.

Renner, I. 2007. Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil.

Roelofsen, J. 2009. Situationsspezifische Prozessplanung und -konfiguration. Flexible Unterstützung in der Produktentwicklung: Prozesse - Daten - Navigation. H. e. a. Meerkamm. Aachen, Shaker: 45-62.

Roelofsen, J. M. K. 2011. Situationsspezifische Planung von Produktentwicklungsprozessen, Verlag Dr. Hut.

Rosemann, M. and J. C. Recker 2006. Context-aware process design: Exploring the extrinsic drivers for process flexibility. The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium, Namur University Press.

Sanchez, R. and J. T. Mahoney 1996. Modularity, flexibility, and knowledge management in product and organization design. Strategic management journal 17(S2): 63-76.

Seol, H., et al. 2007. Design process modularization: concept and algorithm. Concurrent Engineering 15(2): 175-186.

Sered, Y. and Y. Reich 2006. Standardization and modularization driven by minimizing overall process effort. Computer-Aided Design 38(5): 405-416.

Stake, R. 2000. On conceptual development of modular products-Development of supporting tools for the modularisation process, Industriell produktion.

Yassine, A. A., et al. 2013. Simultaneous optimisation of products, processes, and people in development projects. Journal of Engineering Design 24(4): 272-292.

Contact: C. Hollauer, Technical University of Munich, Laboratory of Product Development and Lightweight Design, Boltzmannstraße 15, 85748 Garching bei München, Germany, +49 89/28915136, +49 89/28915144, christoph.hollauer@tum.de