PROCESS OPTIMIZATION BY DSM-BASED MODELLING OF INPUTS AND OUTPUTS

Maik Maurer[1]
(1) Technische Universität München, Germany

ABSTRACT
Robustness is a major challenge for designing engineering processes. And processes are often modeled with Event-driven Process Chains (EPCs). However, the EPCs do not contain sufficient information for analysis and optimization of process robustness. We hypothesize that the quantity of interfaces, i.e. information exchange between tools or organizational units, is an indication for process robustness: The less interfaces occur in a process the more robust it is. And the purposeful interface alignment also improves its robustness. We developed a method for analysis and optimized realignment of process interfaces. We augment EPCs with input-output relations between activities. Then we transfer the process description into a Dependency Structure Matrix and apply a multi-criteria clustering for identifying activity groups, which can be executed without interfaces. The interfaces get assembled between the activity groups, i.e. we define stages for information handover between tools and organizational units. We applied the approach on processes of a mid-sized company and could reliably identify starting points for improving product robustness as well as new layouts of activity groups.

Keywords: process modeling, dependency structure matrix, input-output relations, clustering

1 INTRODUCTION
A company's success is often based upon the quality of its engineering process design. Decisive criteria are e.g. process duration, flexibility and robustness. Several notations and tools have been established for process creation, documentation and mediation of processes to users [1]. Event-driven Process Chains (EPCs) represent a quasi standard notation for (business) process modeling, which is mainly applied in engineering and automotive industry [2]. Basically, EPCs are composed of events, activities, organizational units and tools. The process elements get connected as a process flow. Software tools like ARIS [3] provide users an interface for describing even capacious processes.
Not only product development processes, but also associated organizational, logistic and management processes become increasingly complex [4]. Besides possibilities of modeling, this causes the need for efficient process analysis and optimization. Computer supported (semi-automated) algorithms can be applied for the identification of possibilities for process improvement and for creating suggestions for optimization measurements. EPC modeling provides a large variety of process analyses and optimization approaches. Due to the applied modeling principle, these possibilities are flow oriented. Thus, software for EPC modeling allows users to e.g. identify bottle necks and to optimize the process time by aligning suitable process activities in parallel. However, optimization regarding the information exchange is not considered directly.

2 PROBLEM AND RESEARCH QUESTION
Whereas software supported EPC models allow the description of even huge process flows, this modeling does only provide insufficient information about the process robustness. Most applied robustness indices only consider the quantity of process activities and relations. Some analysis methods even consider the amount of branching points in EPCs. However, this only represents a limited view on process robustness.
One important indication for the robustness of processes is the quantity of existing interfaces (and their level of implementation). Such interfaces can occur between tools, which get applied for subsequent activities within a process. An organizational interface describes the required exchange between people or organizational units, who possess an input-output relation because they execute directly related tasks. Now, one can assume the following:

The less information exchange between people and between tools the more robust the execution of a process gets. As each interface includes the risk of information loss or corruption, fewer interfaces mean a higher security in process execution. The relevant interfaces (tool-based and organizational) can not be determined from EPCs, because this modeling only describes the determined sequence of activities and not their (content-based) input-output relations.
The above statement is clarified in Figure 1, which shows a generic part from an engineering process in EPC modeling. Activity 1 is followed by Activity 2 (passing Event 1). Both activities apply different tools (Tool a, Tool b) and are executed by different people (Person A, Person B). Now, one can assume (and compute) the existence of a tool interface and an organizational interface between both activities. However, this would only be correct, if Activity 1 provides information output, which is required as input for Activity 2. The vertical arrows in Figure 1 do not describe information exchange, thus a statement about the existence of interfaces can not be made. If, for example, Activity 1 represents “Delivery of the product” and Activity 2 represents “Update best-practice documentation”, these activities are only aligned sequentially without any input-output relation. Even if different tools are applied by different people for executing these two tasks, interfaces do not exist between them.

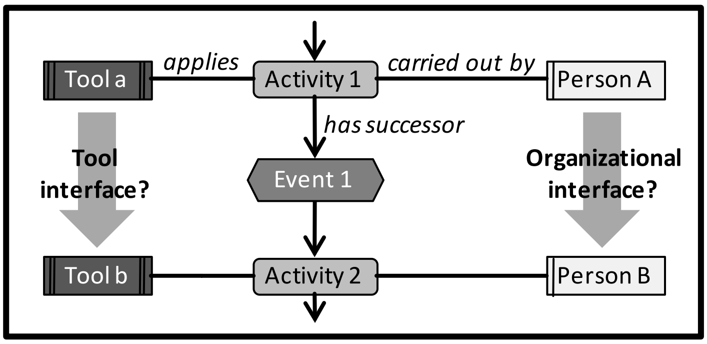


Figure 1. Dependency meaning and interpretation in EPC model

It can be stated that interfaces between tools and involved organizational units contribute significantly to the robustness of a process. EPCs depict the appointed sequence of activities and not information-based input-output relations between them. For this reason, EPCs are not an adequate basis for the analysis and optimization of input-output relations and consequently not for designing processes with a minimized quantity of (potentially harmful) interfaces. This raises the question, how information-based interfaces in processes can be modeled and optimized concerning improved process robustness.

3 OBJECTIVE

We want to augment EPCs with information-based input-output relations between the activities. Then we want to create a process description based on the Dependency Structure Matrix (DSM), which allows to apply clustering algorithms for the identification of an improved alignment of activities. The criteria for clustering can be formulated as follows:

- Accumulation of activities to main tasks if they
    - are linked by input-output relations
    - do not require different tools
    - do not involve different organizational units
- Align activities in different main tasks if they
    - are linked by input-output relations
    - and apply different tools and/or are executed by different organizational units

As for most processes these criteria will not be completely fulfillable, a trade-off has to be found between them. In the end, the process has to be re-transferred into the EPC, because the depiction of the process flow and the designers’ familiarity with the visualization supports the daily work.

4 STATE OF THE ART

4.1 Business process modeling

Typical business process models represent the activity flow [5]. [6] stated that information flows are most important to model, as “information is what flows the most in business processes”. Based on this assumption [7, 8] developed the “communication flow optimization model. In addition, [9] emphasizes the relevance of modeling information flows in process descriptions. In this context, the authors generated 18 categories of information flows. Whereas the importance of information flows has been identified, their adequate optimization in business processes is not systematized so far.

4.2 DSM analysis and optimization

The representation and analysis of processes in matrix notation is established since many years [10]. A summary about available techniques and applications is given by [11]. [12] continues this summary and presents a comprehensive overview of available analysis methods. In contrary to the analysis and optimization of DSM, the management of more holistic system descriptions is not supported adequately. [13] describe the need for combining the three views of component, process, and organizational structures as a basis for successful product development; however, they do not present a solution for this multi-criteria problem. The enhancement of DSM by Multiple-Domain Matrices [4] even highlights the lack of multi-criteria analysis and optimization approaches for matrix notations. Only recently, [14] proposed a procedure for the simultaneous optimization of product, process and people DSM; however, the approach has only been tested on very small examples so far and seems not to be fully developed for application to more complicated system descriptions.

4.3 Acquisition of relations between process elements

The high quality acquisition of dependencies represents a precondition for any analysis and optimization of structural process models. Whereas much research is available for the analysis and optimization of system structures, only few authors concentrate on the acquisition process, e.g. [15, 16]. Most process models represent the aligned sequence of process steps only; often this does not match the structure of inputs and outputs between the process steps. In many cases, such inputs and outputs have to be acquired by interviews with process specialists. Propositions for the systematic acquisition of structural information (system elements and their relations) can e.g. be found in [4].

5 METHODICAL APPROACH

Our approach on process analysis and optimization is subdivided into seven sequential steps. They differ in the required time and effort as well as in available software support. In the steps 1 to 3 the required information is transferred from the EPC model to a matrix notation. Then a condensed DSM is computed, which represents the basis for process analysis, interpretation and optimization. The DSM indicates the information-based input-output relations, which have to pass tool or organizational interfaces. Users can discuss their relevance for the process and their actual quality of implementation. Furthermore, the DSM gets applied for improving the alignment of activities. Finally, the new sequence of activities gets transferred to an EPC.

5.1 Transfer from EPC to matrix notation

For the transfer of the activities from an EPC to a matrix notation the relevant process elements have to be noted as column heads and row heads of a matrix (step 1 in Figure 2). Then the allocation of organizational units and tools to process activities has to be transferred as cell entries (step 2 in Figure 2). In the EPC an allocation is indicated by a connection line between the activity and the tool/organizational unit. In the matrix the same allocation gets represented by a mark in the matrix cell, which connects the row of the activity with the column of the tool/organizational unit. This transfer can be automated, as most EPC modeling tools provide export possibilities.

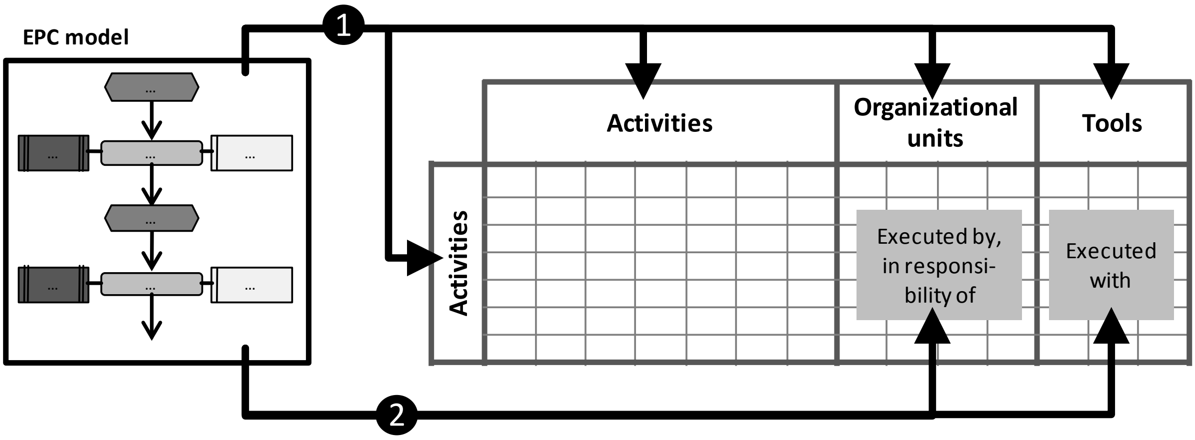


Figure 2. Transfer of process elements and relations from EPC to matrix notation

5.2 Acquisition of input-output relations between activities

The information-based input-output relations between process activities have to be acquired and implemented to the matrix notation. In general, the acquisition could be done by drawing connectors in the EPC followed by an (automated) export to the matrix. However, it is more convenient to directly acquire the relations in the matrix, because EPCs can become very space consuming and implementing additional relations complicates the overview for users. It would become difficult to assure the consideration and acquisition of all possible relations in such a depiction. In contrast to the EPC, the matrix notation is well appropriate (and often applied) for the acquisition of relations [4]. If the activities are noted in the row and column heads (see Figure 3), users can sequentially browse through all matrix cells (representing all possible relations between activities). This acquisition procedure allows maintaining high quality networks; but the acquisition can be time consuming and the support by a workshop moderator is helpful.

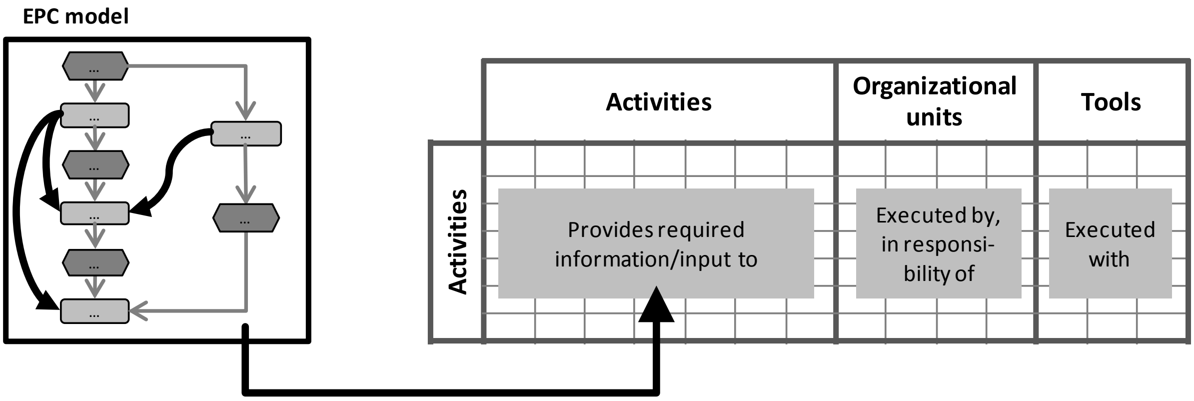


Figure 3. Acquisition of input-output relations between process activities

5.3 Acquisition of existing main tasks

Main tasks are activity groups, which are not defined explicitly. They describe those blocks of activities process designers arrange together within the EPC depiction, because they are related by content. For example, parallel process streams are typically arranged in blocks, which can be easily detected. The information about these blocks has to be transferred to the matrix notation: activities belonging to the same block are aligned together; hereby, the alignment of activities must be identical in the rows and the columns. Consequently, relations between activities of the same block appear only in the matrix cells around the diagonal (see Figure 4).

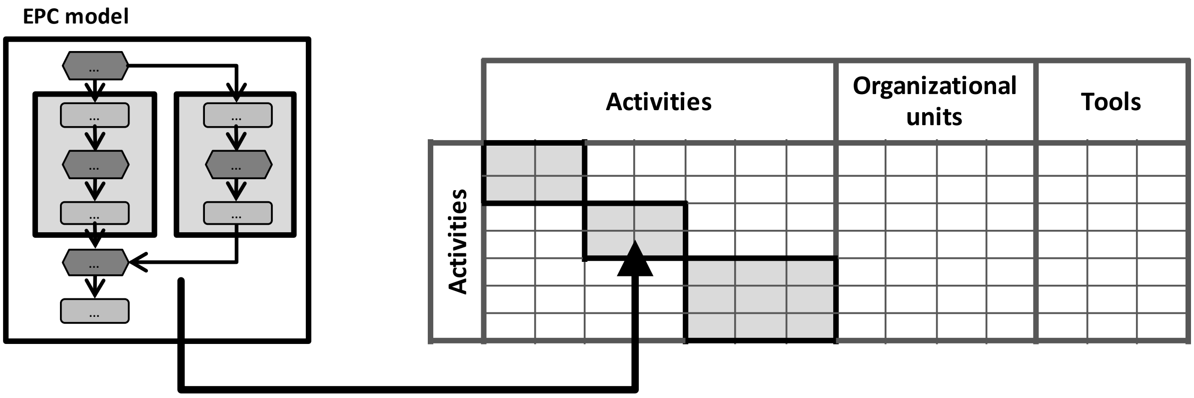


Figure 4. Identification of main tasks within the as-is process

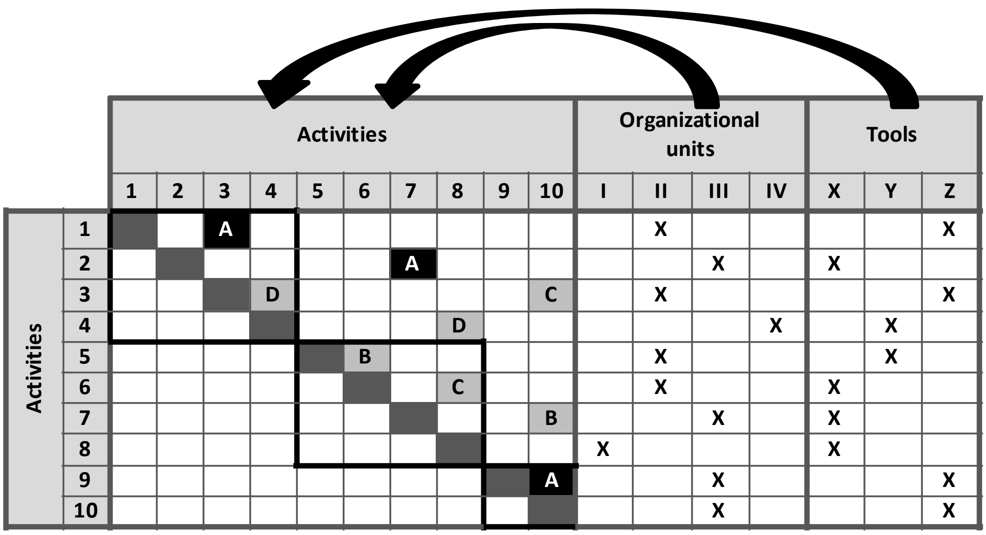
5.4 Computation of the activity DSM

The activity DSM can be computed based on the information implemented to the matrix notation before. The objective is to obtain a DSM containing all acquired input-output relations (see section 5.2) completed with all information-based interfaces between tools and organizational units. These interfaces can be computed as follows:

If two activities are connected by an information-based input-output relation and

- both activities apply identical tools and are executed by identical organizational units: input-output relation without any interface (A)
- the activities apply different tools: input-output relation with tool interface (B)
- the activities are executed by different organizational units: input-output relation with organizational interface (C)
- the activities apply different tools and are executed by different organizational units: input-output relation with tool and organizational interface (D)

In Figure 5 the computation can be retraced. All acquired input-output relations between activities (left part in Figure 5) are indicated with letters A, B, C or D in the corresponding matrix cells. The different letters result from the associated organizational units and tools (right part in Figure 5). For example, Activity 1 and Activity 3 both are executed by the same organizational unit (II) and both apply the same tool (Z). Therefore, the link from Activity 1 to Activity 3 is indicated as A. Activity 7 and Activity 10 also possess an input-output relation, are executed by the same organizational unit but apply different tools. For this reason the relation is indicated as B. Activity 6 and Activity 8 apply the same tool but are executed by different organizational units – which results in the letter C. Finally, the application of different tools by different organizational units results in the indication of an input-output relation with the letter D.



Activities	1	2	3	4	5	6	7	8	9	10	I	II	III	IV	X	Y	Z
1			A									X					X
2							A						X		X		
3				D						C		X					X
4								D						X		X	
5						B						X				X	
6								C				X			X		
7										B			X		X		
8											X				X		
9										A			X				X
10													X				X

Figure 5. Computation of the activity DSM by activity links to organizational units and tools

The determination of this activity DSM can be easily automatized by algorithmic support [4]. Even the computation of activity DSM containing several hundred activities does not require significant computational resources.

5.5 Analysis and interpretation of relevant interfaces

The activity DSM contains information about input-output relations between activity, the existence of interfaces between tools and organizational units as well as the as-is grouping of activities in main tasks. The fundament of the subsequent analysis and interpretation are the following two criteria:

If two activities are connected by an input-output relation and

- are executed by the same organizational unit with the identical tool support, both activities should be aligned in the same activity group (main task)
- are executed by different organizational units or with different tool support, the interface between both activities should be arranged between two activity groups (main tasks).

In other words, activities, which exchange information, should be grouped together if the same people execute them with the same tools. And if different people or different tools are required for delivering information from one activity to another, the interfaces should be concentrated at the changeover from one activity group (main task) to the next one.
Following the hypotheses, Figure 6 indicates four input-output relations, which should be considered for process improvement. Three input-output relations possess organizational or tool-based interfaces, which are integrated in main tasks. One input-output relation connects two activities belonging to different main tasks, but both activities are executed by the same people with the same tool.

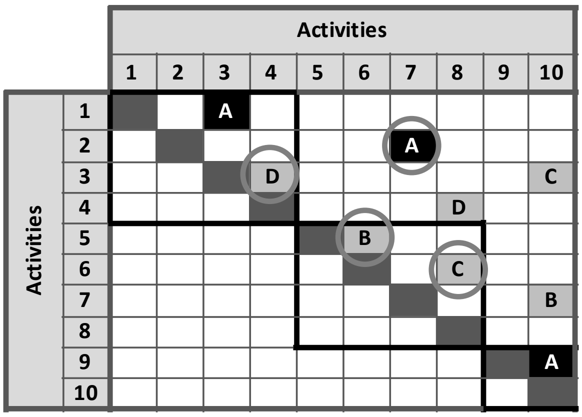


Figure 6. Starting points for process improvement

It must be mentioned that the analysis presented here only considers qualitative information about relations. Further reasons for the as-is-process design are not taken into account. It might be possible that a tool-based interface exists but is perfectly implemented (e.g. by copy-paste procedure). In this case, the interface should not be subject to subsequent process optimization. Interfaces identified by the structural analysis should be rated by process designers concerning their quantitative relevance.

5.6 Propositions for the realignment of process activities

An optimized process design in the meaning of the above multiple criteria (see section 5.5) means: If links between activities are rated by “A”, these links should be included in clusters. At the same time, all links between activities, which are rated by “B”, “C” and “D” should not be part of the clusters.
Clusters can be formed in a DSM by switching the rows and columns [11]. Hereby, the identical order of elements on both axes has to be maintained (simultaneous switching of the rows and columns). A large quantity of clustering algorithms is described in literature [17] and several ones are applied to DSM, e.g. by [18]. However, multi-criteria clustering represents a particular challenge. At this time, appropriate algorithms are not available in the context of DSM application. Anyway, propositions for the realignment of process activities can be generated by manually switching the row and column

alignment in the DSM. If users are familiar with the DSM methodology this can even be executed successfully for DSMs comprising more than 50 activities.
Figure 7 shows an example of an as-is process design at the left side. The arrows indicate the relevant relations, which have to be reordered. Whereas the activities 1 to 4 form a main task, the output generated by Activity 3 is required by a different organizational unit, who executes Activity 4 with a different tool support. Thus, Activity 4 should no longer be included in the same main task. Similar considerations create the need for realigning the activities 5 to 8: In the as-is process this main task includes two interfaces (one organizational, one tool-based interface), which can complicate the process execution. Concerning the relation between Activity 2 and Activity 7, it should be verified, if both activities could be assembled in the same main task (same tool and organizational unit).
The manually realigned matrix is shown at the right side of Figure 7. Here, two main tasks could be identified, both comprising four activities. Within the main tasks only information exchange, but no organizational or tool-based interfaces exist. The practicability and possible constraints have to be rated by process designers afterwards.

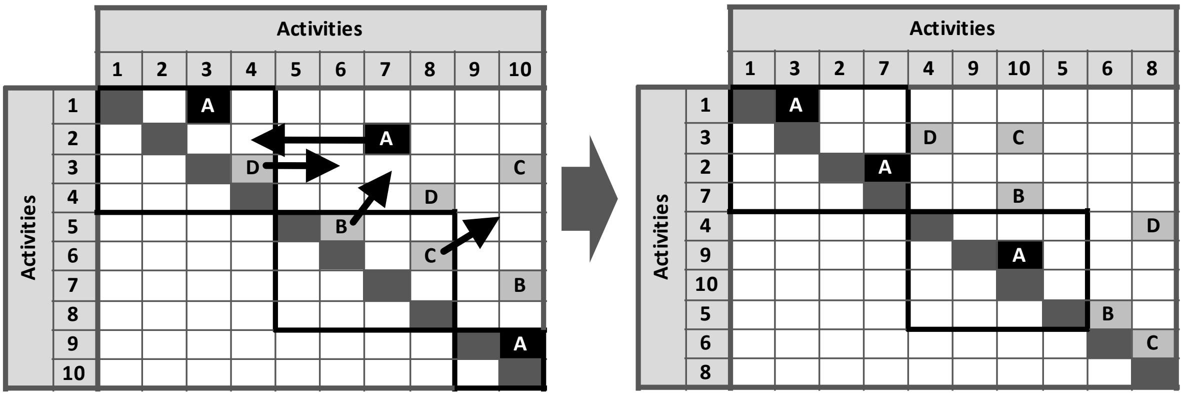


Figure 7. Manual realignment of process activities, multi-criteria clustering

5.7 Transfer from MDM to EPC

The identified optimizations of activity alignment have to be retransferred to an EPC model. This representation is much easier to read and handle in the daily work as a matrix notation. At this stage of development, an automatized transfer procedure can not been provided. The main problem is that not all information from the EPC is represented in the matrix notation. Thus, an automatized re-transfer would result in an incomplete process description. For example, branching points (AND, OR, XOR) are used in EPCs but not in the activity DSM. A manual transfer can be realized as long as the quantity of process activities does not get too large. In the following use case around 50 activities exist. This process size was manageable by manual transfer. Hereby, the improved EPC model was not build up from the scratch, but adaptations were implemented step by step. Nevertheless, a more efficient transfer from matrix notation to EPCs is required in the future.

6 USE CASE: OPTIMIZING THE PROCESS OF PRODUCT PACKING AND SHIPPING

The process considered here has been analyzed in a mid-sized engineering company and describes the packing and shipping of engineered commodities as well as associated administrative tasks.

6.1 Description and analysis of the as-is process

Initially, the process for packing and shipping engineered commodities has been depicted as an EPC model comprising 52 activities. Applied tools and executing organizational units were associated consequentially. The graphical arrangement was easy to understand, as branching points, parallel streams and iterations were clearly pointed out. Nevertheless, the process depiction as an EPC was already too large for analysis and deduction of optimization measures. In printed format, the process requires a sheet in format DIN A0. This makes it difficult for users to get a detailed understanding of the process and to identify shortcomings in the robustness.

Figure 8 shows the activity DSM. Whereas the activities could be extracted from the EPC model, the information-based input-output relations were acquired with an employee of the company. Based on matrix computation (see section 5.4) they were indicated with letters A, B, C and D. This shows the existence of tool and/or organizational interfaces. The framed blocks of activities represent the main tasks, as they were identified in the EPC.

The visual analysis of the matrix notation shows that 13 input-output relations without any tool or organizational changes are located between main tasks (outside of the framed blocks). These relations are indicated by A and black background of the cells. 20 input-output relations with tool and/or organizational interfaces exist within the main tasks, indicated by B, C and D and grey background of the cells.

Three relations are positioned below the diagonal; in general, these represent iterative steps in a process flow. In the use case presented here, we interpret these relations as shortcomings in the original EPC.

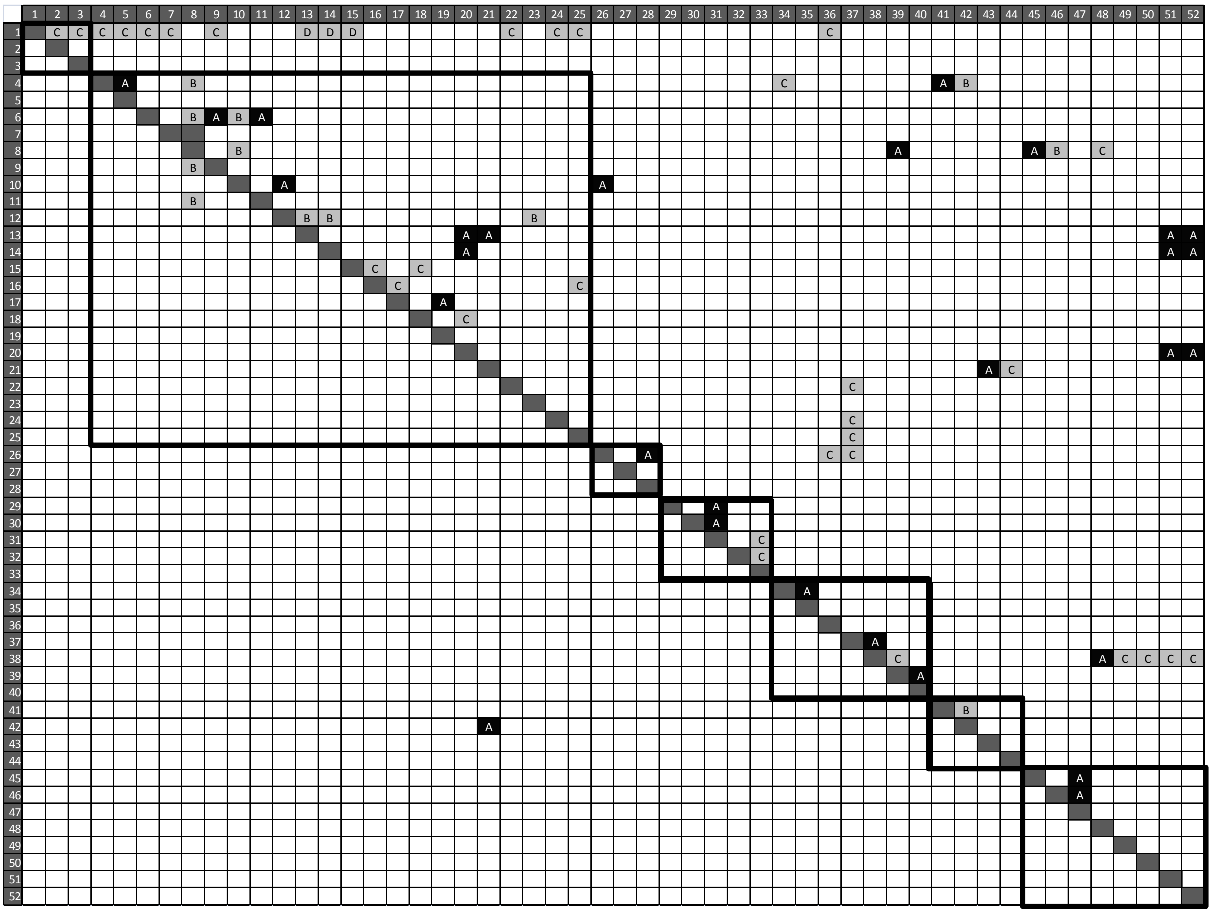


Figure 8. As-is process of packing and shipping engineered commodities in the activity DSM

Before initiating any optimization measures it must be clarified, if the identified concerns represent problems in the real process. For example, if a tool interface is perfectly implemented in the workflow it is not necessary to eliminate it from a main task. Consequently, all shaded matrix cells within the framed blocks as well as all black cells outside of the blocks were discussed with process specialists in the company. However, even if some relations seemed to be more critical than others no interfaces were excluded from possible improvements of the process robustness.

6.2 Optimized Process layout with minimal interfaces

Figure 9 shows an optimized alignment of the process activities. The rows and columns were switched manually according to the criteria mentioned in chapter 3. Even if the process depiction comprises 52 activities, the realignment could be executed within less than one hour. The activities then were aggregated into new activity groups (main tasks), indicated by the framed blocks. Now, only one input-output relation without a tool or organizational interface is located between two main tasks. And only seven tool or organizational interfaces exist within main tasks. That means that in the new process layout the change between applied tools or organizational responsibility is accumulated between the main tasks. This makes the process flow more robust, as potential information loss (due to suboptimal interfaces) is minimized.

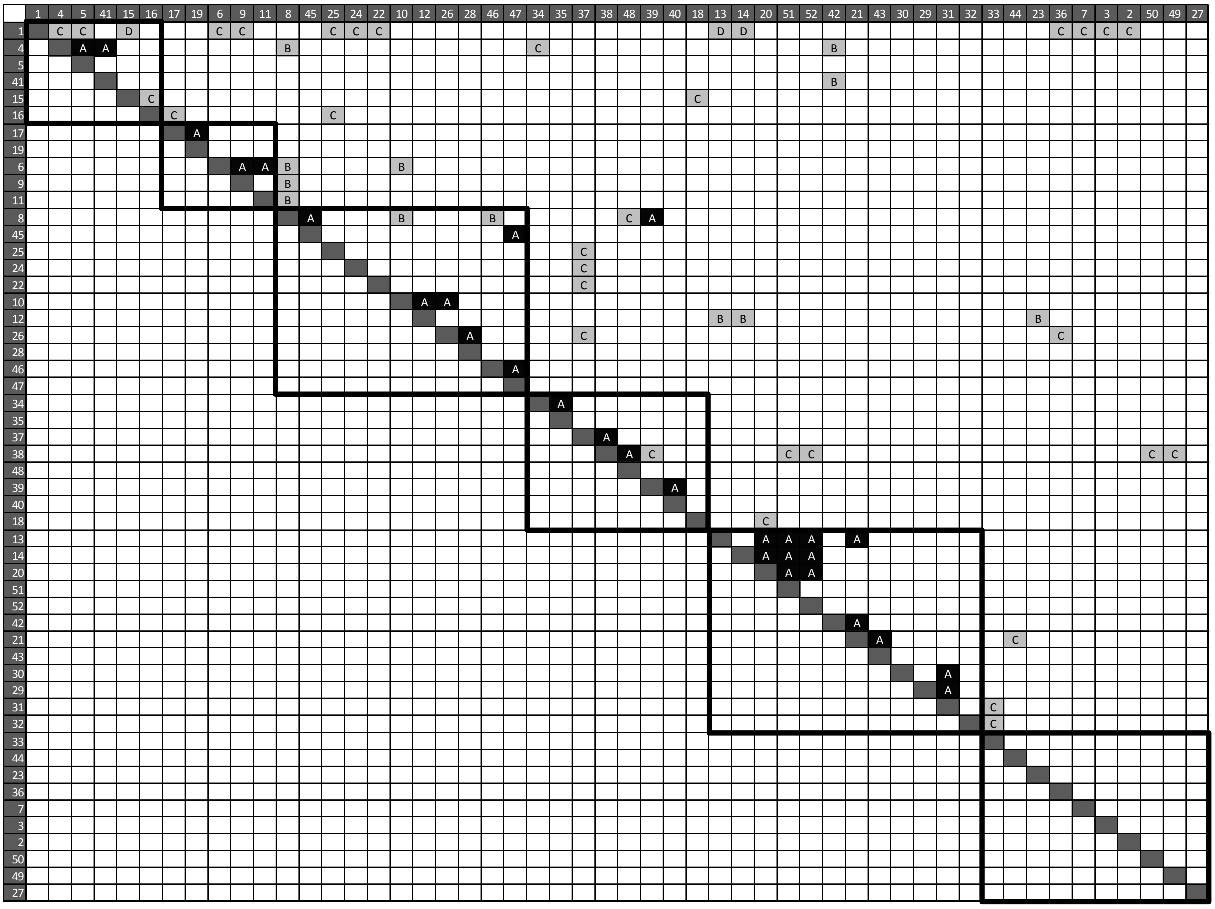


Figure 9. Proposition for an optimized alignment of process activities in the activity DSM

6.3 Highest robustness vs. minimal run-time

Even if the optimization presented in Figure 9 considers multiple criteria (clustering of purely information-based relations, alignment of tool and organizational interfaces between clusters) it is still limited to the improvement of interfaces only. Besides the process robustness, however, process runtime represents a very important issue. A minimum process runtime can be achieved by consequently parallelize process activities, which can be executed independently from each other. The approach presented here applies the input-output relations between the activities and therefore can serve the information needs for parallelization. However, the parallelization of activities downgrades the quality of interface alignment. That means that process robustness (interface positioning) and process run-time (parallelization of activities) represent conflicting objectives for the process optimization. In the use case presented here, the process owners set the value on process robustness; therefore the optimization of interface alignment was beneficial.

7 CONCLUSION AND FUTURE WORK

The presented approach on process optimization helps improving the process robustness by identifying and realigning the tool and organizational interfaces between activities. The activities are extracted from EPCs. Input-output relations between the activities have to be acquired, because EPCs only show the defined flow sequence (and not the information delivery/needs).

An optimized alignment of process activities has to be done manually and can not be automatized so far, because the optimization challenge represents a problem of multi-criteria clustering; appropriate algorithms are not available for application to DSMs.

Processes possess branching points and different sequences of activities. Depending on the specific process, a path can be run through more often than another. As a consequence, interfaces in a process can have different importance, just because they appear more often in practice. Assumed that such quantitative information about the process flow is known, this would represent a further criterion for the multi-criteria optimization.

Improvement in process runtime represents another optimization objective. This is not considered by the approach presented here. Unfortunately, improved process runtime often means disadvantageous alignment of interfaces, which results in decreasing process robustness. That means that an optimized process layout means to find a trade-off between minimum runtime and highest robustness.

In summary, the approach presented here represents a systematic procedure for improving the robustness of a company's engineering and business processes. Its application on processes in a mid-sized company led to significant results while the complexity of application was manageable. Future research will focus on algorithms for the multi-criteria clustering.

REFERENCES

[1] Browning T.R. and Ramasesh R.V. A Survey of Activity Network-Based Process Models for Managing Product Development Projects. *Production and Operations Management*, 2007, 16(2), 217-240.

[2] Spath D. and Weisbecker A. *Business Process Management Tools 2008*, 2008 (Fraunhofer IRB, Stuttgart)

[3] Scheer A.-W. *ARIS - Business Process Modeling*, 1999 (Springer, Berlin).

[4] Lindemann U., Maurer M., Braun T. *Structural Complexity Management – An Approach for the Field of Product Design*, 2009 (Springer, Berlin).

[5] Hammer M. Process Management and the Future of Six Sigma. *MIT Sloan Management Review*, Winter 2002.

[6] Kock N., Murphy F. *Redesigning Acquisition Processes: A new methodology based on the flow of knowledge and information*, 2001 (Defense Acquisition University Press, Fort Belvoir, Virginia).

[7] Kock N., Communication-focused business process redesign: Assessing a Communication Flow Optimization Model Through an Action Research Study at a Defense Contractor. *IEEE Transactions on Professional Communication*, 2003, 46(1), 35–54.

[8] Kock N., Danesh A., Komiak P. A discussion and test of a communication flow optimization approach for business process redesign. *Knowledge and Process Management*, 2008, 15(1), 72-85.

[9] Berente N., Vandenbosch B., Aubert B. Information flows and business process integration. *Business Process Management Journal*, 2009, 15(1), 119-141.

[10] Steward D. The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transaction on Engineering Management* 1981 28(3), 79-83.

[11] Browning T. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. *IEEE Transactions on Engineering Management* 2001 48(3), 292-306.

[12] Maurer M. *Structural Awareness in Complex Product Design*. 2007 (Dr. Hut, München), online: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20070618-622288-1-1

[13] Eppinger S.D. and Salminen V.K. Patterns of product development interactions. In *International Conference on Engineering Design, ICED'01*, Vol. 1, Glasgow, August 2001, pp.283-290 (Professional Engineering Publishing, Bury St Edmunds).

[14] Yassine A. Multi-Domain DSM: Simultaneous Optimization of Product, Process & People DSMs, In *12th International DSM Conference*, Cambridge, UK, July 2010

[15] Dong Q. *Predicting and managing system interactions at early phase of the product development process*. Ph.D. Thesis, 2002 (MIT, Cambridge USA).

[16] Bartolomei J.E. *Qualitative knowledge construction for engineering systems: extending the design structure matrix methodology in scope and procedure*. Ph.D. Thesis, 2007 (MIT, Cambridge USA).

[17] Hartigan J.A. *Clustering Algorithms*. 1975 (John Wiley & Sons, New York).

[18] Kusiak A. *Engineering Design – Products, Processes and Systems*. 1999 (Academic Press, San Diego).

Contact: Dr.-Ing. Maik Maurer
Technische Universität München
Institute of Product Development
Boltzmannstr. 15
85748 Garching, Germany
Tel: Int +49 89 289 15133
Fax: Int +49 89 289 15144
Email: maik.maurer@pe.mw.tum.de
URL: http://www.pe.mw.tum.de

Dr.-Ing. Maik Maurer teaches complexity management at the Technischen Universität München, Germany and is in charge of several research projects in the fields of systems engineering and complexity management. He is co-founder and CEO of the company TESEON, providing software and consultancy for the management of complex systems.