INTERNATIONAL DESIGN CONFERENCE - DESIGN 2018
https://doi.org/10.21278/idc.2018.0434

AUGMENTED DSM SEQUENCING TO SUPPORT PRODUCT DEVELOPMENT PLANNING

D. Göhlich, S. Hildebrand and D. D. Schellert

Abstract

The Sequencing of product DSMs has been proposed for the identification of interdependencies and for product development planning (PDP) including components and subsystems. But, due to little exploitable outcome, this approach is hardly used in practice. A new method using an augmented DSM Sequencing algorithm with a gradient based optimizer is presented. The method considers Product Architecture Drivers to derive a suggestion for an efficient planning process and is integrated in a DSM software. Furthermore, it is validated using a combustion engine as a typical engineering product.

Keywords: product development, design methods, design structure matrix (DSM), augmented sequencing algorithm

1. Introduction

Engineering products become increasingly complex and the markets of the future are expected to become even more competitive than today. Hence, efforts for product development shall be minimized and an efficient planning of the product development processes (PDP) is a decisive success factor (Lindemann, 2016a). One of the major challenges of time and cost effective PDP is the identification and the management of interdependencies of subsystems and components. This task can be supported by using a Design Structure Matrix (DSM) (Steward, 1981).

The DSM concept offers many powerful and to many different systems generically applicable tools (Eppinger and Browning, 2012), however, after a first utilization in industry in the 1990ies, today it seems to be hardly used in practice (Eppinger and Browning, 2012). Possible reasons are the required amount of data and the difficulty to obtain and organize them (Browning, 2001). By the use of today's computing capacity, the available data can - also for larger matrices - be evaluated more in-depth and by means of advanced statistic approaches. By the computer-aided, statistic evaluation of big amounts of data, also the creation of big DSMs can be automated and thus be sped up (Gopsill et al., 2017). In contrast the investigation of new (sequencing) algorithms has found very little research attention.

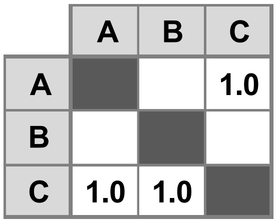
In this paper, a new method based on an augmented DSM sequencing algorithm and in consideration of Product Architecture (PA)-Drivers is presented, to derive a suggestion for an efficient product development planning from a given product architecture. The method is validated on the basis of a CAD-model for multi-body (CAD-MBS) model for engine simulation provided by the CONTECS engineering services GmbH, with the aid of the specifically developed DSM-Software MPM Pro.

2. Applied methods

2.1. Design Structure Matrix

A Design Structure Matrix (DSM) represents the dependencies between elements of one single type in a system (Kreimeyer and Lindemann, 2011). Depending on the type of the elements, DSMs are compiled "process oriented" (e.g. working process, development tasks) or "product oriented" (e.g. components, assemblies, etc.).

A DSM is always square, since the elements are enumerated in the same order into rows and columns. The matrix entries represent the dependencies and are numerically weighted below. As reading direction is declared: "Element in column influences element in line" (s. Figure 1).



	A	B	C
A	■		1.0
B		■	
C	1.0	1.0	■

Figure 1. Simplified DSM; in this example, element C is dependent on element B (B influences C), but B is not dependent from C

To expand the DSM for multiple domains, "Domain Mapping Matrices" (DMM) and "Multiple Domain Matrices" (MDM) were developed, which can represent systems with multiple hierarchy levels and multiple element types (Bartolomei et al., 2007; Maurer, 2007; Kreimeyer and Lindemann, 2011).

2.2. Sequencing

DSM sequencing is an analysis technique mainly for task or process oriented DSMs (Eppinger and Browning, 2012) to visualize the information flow between several development activities. Therefore, the elements in the DSM are reordered so that ideally every element only depends from its predecessors (Kreimeyer and Lindemann, 2011) i.e. the topological ordering of a graph, of which the DSM is the adjacency matrix (Kreimeyer and Lindemann, 2011, p. 48).

In practice, this is not always possible due to the fact that often there are cyclic dependencies, i.e. not all elements can be sorted into a series (Kreimeyer and Lindemann, 2011; Eppinger and Browning, 2012) Then, the elements that cyclically influence each other must be merged into a so called "partition" which indicates a development iteration loop (Lindemann, 2016b).

In the next step, the remaining elements are reordered using a topological sort. Finally, the partitions are expanded to the contained elements. Note that the elements within the partitions have not been reordered meaningfully since no further information about the relations within a partition is evaluated.

It should be noted, that consecutive elements without dependencies between each other can be handled in parallel or in arbitrary order. Therefore, several solutions may exist, from which one is found by the algorithm.

For the visual evaluation of the sequencing results, several approaches can be found in the literature (Steward, 1981; Browning, 2001; Yassine, 2004; Bartolomei et al., 2007).

2.3. Product architecture and PA-Drivers

The product architecture describes the coherence between the function structure of a product, that describes the product on a functional level, and its physical configuration (Pahl and Beitz, 2013).

Product architecture Drivers (in short: PA-Drivers) are a concept to plan product structures and was developed independently from the DSM concept. As PA-Drivers are called these components of a system, irrespective of their complexity level, that - due to their properties or functions - have a distinctive influence on the product architecture and many other components.

They comprise the functional specifications, geometric parameters of modules and components which have significant impact on the internal and external variance of the product. (Thumm and Göhlich, 2015)

Typically, the number of PA-Drivers is significantly lower than the number of components in a product, and conceptual design of PA drivers should be specified at the beginning of the development process. (Thumm et al., 2014)

To identify PA-Drivers, the product architecture has to be analysed regarding three main perspectives (Thumm and Göhlich, 2015):

1. Functional perspective: Represents the requirements to the product and leads to the main functional specifications
2. Performance perspective: Represents further functional specifications and technical feasibility which lead to the main technical parameters
3. Structural perspective: Includes an outlook onto the expected physical implementation and module/ component structure

3. State of the art and need for research

The DSM concept has been analysed regarding its use in development planning, usually focusing on process DSMs which are generated by decomposing the development processes rather than the product that has to be developed (Danilovic and Browning, 2007). This is unsatisfactory, since the product development is governed by the product architecture as discussed in Section 4.

DSM sequencing tries to find an appropriate order of processes where the number of information feedback loops is minimal. Therefore, interdependent tasks are grouped into partitions and the remaining elements are put in a sequence, where the number of entries above the main diagonal is minimal (Steward, 1981; Bartolomei et al., 2007; Kreimeyer and Lindemann, 2011). This procedure is the same as the topological sort of the digraph corresponding to the DSM (Hildebrand, 2015).

For elements within a partition, the conventional DSM sequencing provides no further information for a reasonable order and it is supposed that all these elements have to be handled virtually simultaneously with tight coordination. Known approaches to deal with partitions, simply acknowledge the existence of partitions without special treatment (Eppinger and Browning, 2012), while others, such as tearing, rely on the removal of dependency information from the matrix based on user decisions und thus imply the manipulation of the decomposition results represented by the DSMs (Steward, 1981; Yassine, 1999; Austin, 2000), an approach critically questioned in the literature (Bartolomei et al., 2007).

In practice, within a complex development iteration loop with many elements, smaller iterations with less elements are to be expected, i.e. nested loops which can be perceived as subpartitions. The conventional DSM sequencing does not provide any insight about the existence and arrangement of such subpartitions. All the identified partitions are on the same hierarchy level, no subpartitions can be identified.

This is especially problematic when only a few, very large partitions emerge from the sequencing analysis as it often happens in industry grade applications according to the authors' experience. A literature review supports this finding (Thebeau, 2001; Eppinger and Browning, 2012).

Hence, the conventional DSM sequencing is not useful for rather complex and strongly connected systems, since in these cases usually multiple large partitions emerge.

In the article at hand, an augmented sequencing algorithm is presented, which addresses the ordering within big partitions. It also regards additional information about the product architecture in the form of PA-Drivers to find an optimized order for all elements in process oriented DSMs.

This algorithm is embedded in a method to derive a product development plan from a product architecture. Finally, this method is validated using a combustion engine as an example.

4. Approach

The approach presented below works with a product architecture and a product DSM as input. With supplementary information, a process DSM is generated and then evaluated to support product development planning. To handle the DSMs and to conduct the evaluation, the software MPM Pro is implemented.

4.1. Methodical procedure

A central element of the methodical procedure is the change of perspective from a product to a process point of view as shown in Figure 2.

Based on an existing product architecture with stated PA Drivers and with the support of a domain expert, product DSMs in multiple hierarchy levels are generated. Therefore, the relations between the components are determined separately, distinguished between geometric und functional dependencies, to maintain the distinction between the function structure and the physical configuration in the sense of the product architecture. Furthermore, the granularity of the obtained information is refined and an evaluation for different scopes of application is made more suitable. For instance, the geometric dependencies can be more relevant for design-engineering aspects, while the functional dependencies are preferable for dimensioning. The selection of the dependency types in use may vary for different applications (Pimmler and Eppinger, 1994; De Weck, 2007; Eppinger and Browning, 2012).

To derive a product development plan, it is necessary to gather the development tasks with their required flow of information (Schellert, 2014). To estimate these pieces of information in an early phase of the development planning, we suggest to consider the already assembled product DSMs as process DSMs by interpreting the product components that have to be developed as development tasks (s. Figure 2). Functional and geometric dependencies are equally considered (Schellert, 2014).

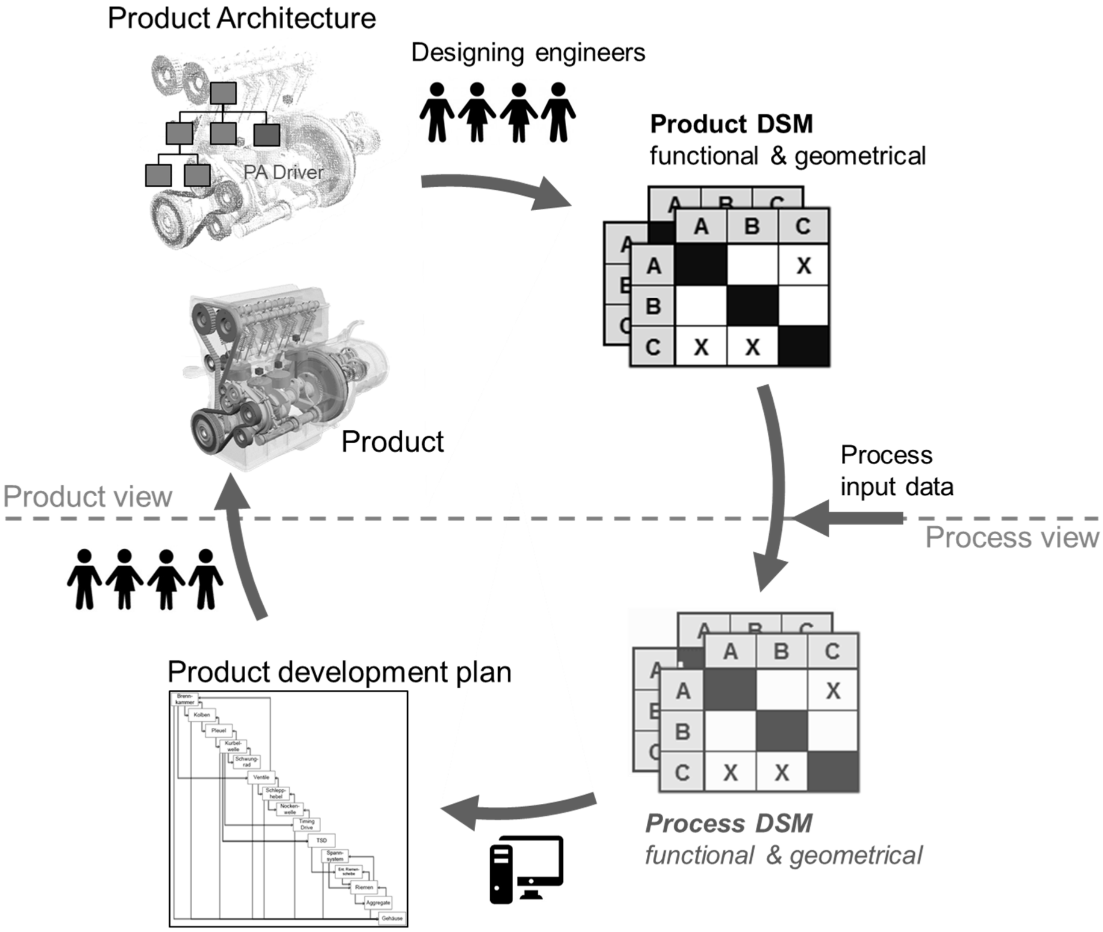


Figure 2. Methodical procedure

More accurate results can be obtained if additional process input data is considered. For example, preparational and follow-up activities can be included in the Process DSMs as separate elements and a distinction between conceptional, design and elaboration can be made. During a development project, the DSMs have to be permanently updated. After each update, the following analysing procedure must be executed again and the obtained results have to be re-evaluated.

4.2. Augmented sequencing

Subsequently, an augmented Sequencing is applied onto the process DSMs to generate a more accurate overview over possible process orders. The standard DSM sequencing algorithm (after Gebala and Eppinger, 1991) which implements the idea of a topological sort is augmented with a gradient based optimizer using a scalar rating function.

The rating function introduced below takes two circumstances into account:

1. Not only the number of iterations, but also the number of components involved in development iteration loops shall be minimized. In a sequenced DSM, a rebound to an earlier development step is described by an entry above the main diagonal. The larger the distance between such an entry and the main diagonal, the more elements reside between the start and the end element of the rebound, i.e. the more elements are involved in the corresponding development iteration loop (Eppinger and Browning, 2012).
2. PA-Drivers should be handled as early as possible in the development process (Hildebrand, 2015), because PA-Drivers influence the Product Architecture significantly (Thumm and Göhlich, 2015). That's why changes on PA-Drivers mostly involve a particularly extensive effort for adaptations at other components.

Hence, a rating function is introduced, whose value becomes minimal for an optimal sequencing order. Punished are DSM entries above the main diagonal, as well as the position of PA-Drivers, which are not put in order at the beginning of a partition. The rating function is constructed as following:

$$sequenceVal := \sum_{i,j} distance(i,j) \cdot DSM_{i,j} + \sum_{Partitions,PA-Driver} (Index_{PA-Driver}) \quad (1)$$

The distance to the diagonal is calculated by:

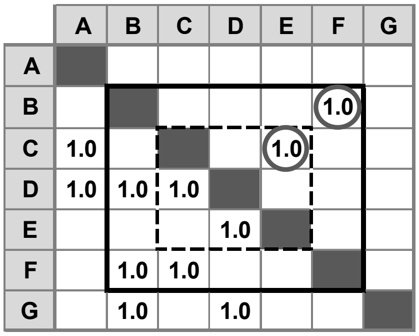
$$distance(col, row) := \max\left\{0, \frac{(col-row)}{\sqrt{2}}\right\} \quad (2)$$

$Index_{PA-Driver}$ is the position of a PA-Driver element within the partition, where the PA-Driver is located. If a PA-Driver is not located in a partition, it holds: $Index_{PA-Driver} = 0$

The value of the rating function increases, the further entries reside above the main diagonal and the stronger these dependencies are. Furthermore, it increases, if PA-Driver are not located at the beginning of partitions.

The optimization happens by a systematic variation of the element order within the partitions. Repeatedly two elements within the same partition are so selected and then pairwise swapped in their positions, that $sequenceVal$ decreases as much as possible, until a further reduction is not possible anymore.

After the optimization, subpartitions can be identified by considering every entry above the diagonal, which is located the furthest right in its row and the furthest up in its column, respectively, as the upper right corner of a square. The upper left corner of this square is the entry on the main diagonal, which is located in the same row, as the entry that causes the kickback. The lower right corner is the entry on the main diagonal, which is located in the same column, as the entry that causes the kickback (s.a. Figure 3). All elements, which are located in the lines (or columns, resp.) between the boundaries of this square, are contained in the subpartition. All dependencies, which are represented by entries within the square, have to be regarded in the development iteration loop described by the subpartition.



	A	B	C	D	E	F	G
A	■						
B		■				1.0	
C	1.0		■		1.0		
D	1.0	1.0	1.0	■			
E				1.0	■		
F		1.0	1.0			■	
G		1.0		1.0			■

Figure 3. Identification of subpartitions

This identification of subpartitions now yields meaningful results, because the elements within the partitions are reordered by the optimizer. Otherwise, the groups of elements which are identified as subpartitions would depend on the order of elements in partitions which originates from the manual creation of the DSM during the product architecture analysis.

4.3. Software

To test the advanced DSM techniques and to validate the presented method, the standalone DSM and MDM software MPM Pro was developed further from an earlier version (Hildebrand, 2015). With MPM Pro, DSMs, DMMs and MDMs can be created and filled in with various dependency types and additional information supported by a rich graphical user interface. The developed software also allows the assignation and further consideration of PA-Drivers. In addition, DSM techniques such as clustering (Gutierrez Fernandez, 1998; Thebeau, 2001) and sequencing can be carried out, each in a classical variant as found in the literature and in several extended versions. For evaluation, a graph representation is available in addition to the matrix view. The interoperability with other programs is given by numerous interfaces (CSV, LaTeX, PNG, XML).

4.4. Application of augmented DSM sequencing to support product development planning

Since the augmented sequencing generated a meaningful order of all elements in a process oriented DSM, within and outside of partitions, it becomes possible to use this order as a suggestion for the handling of the tasks depicted by the DSM. This suggestion might be adapted by the development management or the developers themselves, but it is a starting point for further planning.

To clearly represent the findings from the sequencing, MPM Pro generates a new, graph-like waterfall view of the DSM. This view contains exactly the same information as the DSM, but in a different, more user-friendly manner which emphasizes the element order obtained by the sequencing. Information about the coupling of tasks into partitions, sequential tasks and possibly parallel tasks can be recognized in both representations.

The difference between the waterfall view and an engineering development plan are the missing estimates for the duration of each task. This can be ruled out by making an estimate for each element while creating the DSM or applying these estimates at the end to the waterfall view.

5. Validation using a combustion engine

The presented approach is applied to development of a combustion engine in the early engineering phase, where preliminary CAD models are derived for static, fatigue, thermal and dynamics simulation. Figure 4 shows such model for multi-body engine simulation (CAD-MBS). The transmission and the balance shaft are neglected for further analysis because they only have one dependent interface to the remaining components and therefore do not have a significant impact on the product architecture.

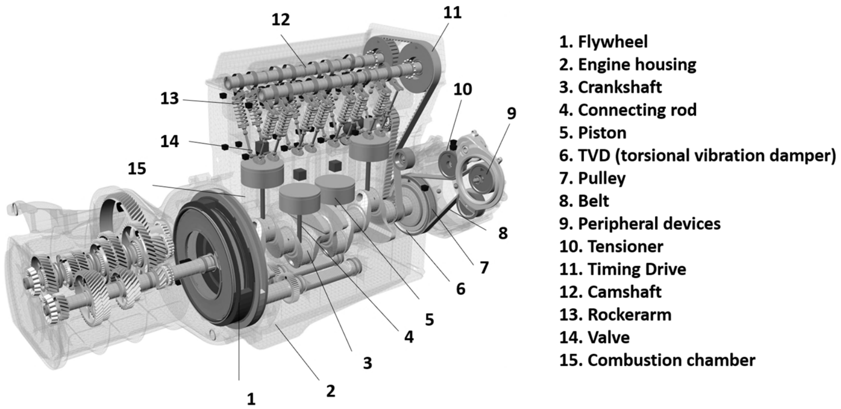


Figure 4. Engine model, courtesy of CONTECS engineering services GmbH

The model was decomposed with the help of a domain expert into hierarchically structured domains (modules, components and parts) and was represented in form of a MDM. Here only the DSM at component level is considered, but functional and geometric dependencies were analysed and represented in different DSMs (s. Figure 5).

Applying the method described in 2.3., the combustion chamber was identified as a PA-Driver by the domain expert. The combustion chamber subsumes the basic requirements of the engine as well as the development results obtained by the thermodynamic and combustion experts before the mechanical simulation of the engine can be conducted.

DSM Components func. dependencies	Crank.	Fly.	Eng. h.	TVD	Pulley	Belt	Per. d.	Tens.	Cam.	Valve	Rock.	Tim. d.	Com.	Con. r.	Piston
Crankshaft	■	1.0												1.0	
Flywheel	1.0	■													
Eng. Housing	1.0		■				1.0	1.0	1.0	1.0	1.0	1.0	1.0		1.0
TVD	1.0			■											
Pulley				1.0	■	1.0									
Belt					1.0	■	1.0	1.0							
Peripheral devices (belt)						1.0	■								
Tensioner (belt)						1.0		■							
Camshaft									■		1.0	0.4			
Valve										■	1.0		1.0		
Rockerarm									1.0	1.0	■				
Timing Drive (Kettentrieb)									1.0			■			
Combustion chamber										1.0			■		1.0
Connecting rod.	1.0													■	1.0
Piston													1.0	1.0	■

DSM Components geo. depencencies	Crank.	Fly.	Eng. h.	TVD	Pulley	Belt	Per. d.	Tens.	Cam.	Valve	Rock.	Tim. d.	Com.	Con. r.	Piston
Crankshaft	■	1.0												1.0	0.7
Flywheel	1.0	■													
Eng. Housing	1.0	0.7	■				0.7	1.0	1.0	1.0	1.0	1.0	1.0	0.7	1.0
TVD	1.0			■											
Pulley	1.0			0.7	■	1.0									
Belt					1.0	■	1.0								
Peripheral devices (belt)						1.0	■								
Tensioner (belt)						1.0		■							
Camshaft									■						
Valve										■	1.0		1.0		0.7
Rockerarm									1.0		■				
Timing Drive (Kettentrieb)									1.0			■			
Combustion chamber										1.0			■		1.0
Connecting rod.	1.0													■	1.0
Piston	0.7									0.7			1.0	1.0	■

Figure 5. Func. dependencies DSM (left) and geo. dependencies DSM (right)

For further analysis, functional and geometric dependencies can be evaluated individually, in sum or as the maximum value of both dependency types. This decision should be made depending on the application, e.g. for the coordination of interfaces and components within assemblies the analysis can be limited to the geometric dependencies. In order to take all dependency types into account, the maximum of the values for both dependency types should be considered. In this article the maximum value is chosen for the analysis to gain an understanding for the engine model in respect of both the functional and the geometric perspective.

5.1. Evaluation of the results

5.1.1. Conduction of the sequencing algorithms

In a first step, the conventional DSM sequencing algorithm is applied to the component DSM. The result is shown in Figure 6 (left). Secondly, the augmented sequencing algorithm is applied to the component DSM, s. Figure 6 (right). For a clearer representation, only subpartitions with at least three elements are marked.

DSM Components max dependencies conventional seq.	Piston	Cran.	Flyw.	***Com. c.***	Cams.	Valve	Rock.	Tim. d.	Con. r.	TVD	Tens.	Pulley	Belt	Per. d.	Eng. h.
Piston	■	0.7		1.0		0.7			1.0						
Crankshaft	0.7	■	1.0						1.0						
Flywheel		1.0	■												
Combustion chamber	1.0			■		1.0									
Camshaft					■		1.0	0.4							
Valve	0.7			1.0		■	1.0								
Rockerarm					1.0	1.0	■								
Timing Drive					1.0			■							
Connecting rod	1.0	1.0							■						
TVD		1.0								■					
Tensioner											■		1.0		
Pulley		1.0								1.0		■	1.0		
Belt											1.0	1.0	■	1.0	
Peripherical devices													1.0	■	
Engine housing	1.0	1.0	0.7	1.0	1.0	1.0	1.0	1.0	0.7		1.0				■

DSM Components max dependencies augmented seq.	***Com. c.***	Valve	Piston	Con. r.	Crank.	Valve	Flyw.	Rocker.	Cams.	TVD	Tens.	Pulley	Belt	Per. d.	Eng. h.
Combustion chamber	■	1.0	1.0												
Valve	1.0	■	0.7				1.0								
Piston	1.0	0.7	■	1.0	0.7										
Connecting rod			1.0	■	1.0										
Crankshaft			0.7	1.0	■	1.0									
Flywheel					1.0	■									
Rockerarm		1.0					■	1.0							
Camshaft							1.0	■	0.4						
Timing drive								1.0	■						
TVD					1.0					■					
Tensioner											■		1.0		
Pulley					1.0					1.0		■	1.0		
Belt											1.0	1.0	■	1.0	
Peripherical devices													1.0	■	
Engine housing	1.0	1.0	1.0	0.7	1.0	0.7	1.0	1.0	1.0		1.0			1.0	■

Figure 6. Results of conventional sequencing (left) and augmented sequencing (right)

5.1.2. Comparison of the results from the conventional and the augmented sequencing algorithm

In order to evaluate the augmented sequencing, the results of the conventional and augmented sequencing are compared. Both sequencing approaches deliver a DSM that is divided into two big partitions. The TVD and the housing are outside of both partitions and therefor treated equally in both cases.

In the conventional sequencing the order of the elements within a partition is not very meaningful because it is random and does not convey any additional information. For example, the combustion chamber is located in the middle of the first partition after the flywheel, while the piston is the very first element in the DSM.

Moreover, no subpartitions can be made out because the number and layout of entries above the main diagonal within the partitions is by chance thus allowing no further conclusions. For the practical use, a partition as large as the first one generated by the conventional sequencing does not convey any new or useful information.

The augmented sequencing provides the same partitions, but the elements within the partitions are now ordered by the optimization (see Figure 6) providing that the number of entries above the diagonal even within the partitions is reduced (Steward, 1981; Gebala and Eppinger, 1991). The PA-Driver combustion chamber is set as the first element in the partition.

Thus, a statement about a reasonable order of the elements within the partition can be made and a better overview about the inevitable kickbacks can be obtained. In addition, the partitions can be divided into smaller subpartitions, as explained above, to improve the overview for the spectator and widen the system understanding for the analysed engine model.

5.1.3. Evaluation of the augmented DSM sequencing

To evaluate the results of the augmented sequencing, the generated waterfall view is compared to the reference development process of combustion engines. The reference development process in Figure 7 (left) is the standard approach in an early development phase used by CONTECS engineers and has evolved over many generations of engine development. The arrangement of elements in groups represent coupled components that have to be developed simultaneously. Within each group, no dependencies between the components are specified. Connections between elements represent the chronological order for developing the components and do not represent functional dependencies between them.

In Figure 7 (right) the waterfall view as generated by the augmented DSM sequencing is shown. In comparison, a significant similarity between both representations can be recognized. The order of elements is roughly equal in both representations, while the computer-generated waterfall view contains more information such as iteration loops and dependencies within partitions (s. Figure 7, right). Moreover, the layout of coupled, simultaneously developed elements is in agreement with the reference process.

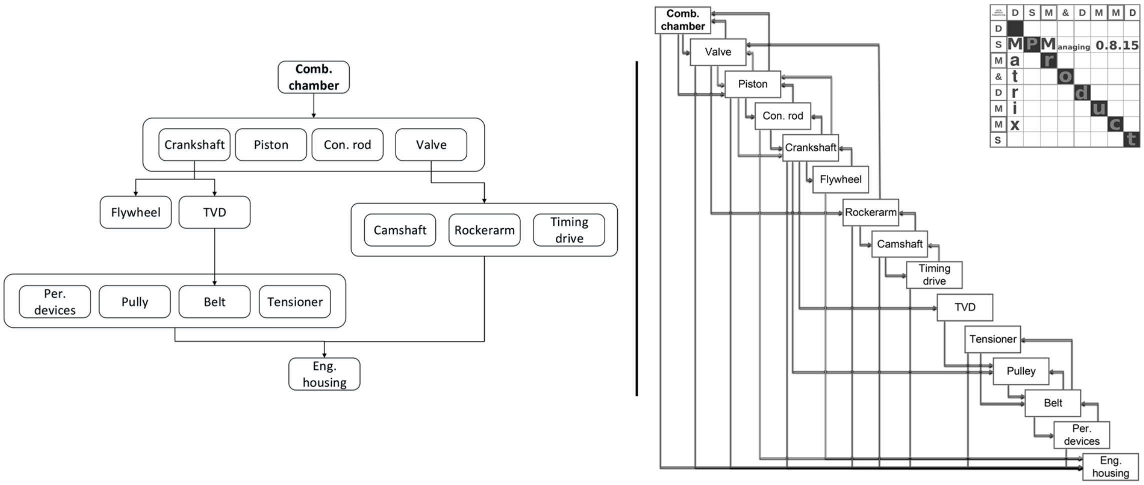


Figure 7. Reference process (left) and computer-generated waterfall view of augmented DSM sequencing results (right)

The rockerarm has bidirectional dependencies with the valve, for that reason the sequencing counts the components of the chain drive (camshaft, rockerarm and timing drive) to the first partition. The augmented sequencing, however, recognizes these subpartition at the end of the first partition, which comes very close to the practical development process.

6. Conclusions and future work

Through the decomposition of a product and its product development process profound system knowledge can be built up. To achieve this, a basic understanding of the examined system is required. This basic understanding can be achieved through the questioning of several design engineers specialized in certain components and without a comprehensive overview of the complete product.

With the acquired system understanding, information of the product architecture expressed via PA-Drivers and the detailed system decomposition, a product DSM is generated, then transformed into a process DSM which can be analysed by the augmented sequencing algorithm to determine a well-suited development process. The information gain mainly serves the purpose of technical development planning and also provides clues to the work organization of the specialists.

However, the system knowledge gained in the data collection must be considered in the evaluation of the results in order to detect and compensate any possible deficiencies of the used DSM techniques. A detailed control of the result by the user is therefore required. For example, in the considered DSM matrix, the dependency of the TVD is only correctly mapped by a rough design of the crankshaft. As a remedy, the rough and detailed design of the crankshaft can be seen as two individual work processes with corresponding dependencies.

Applying a conventional DSM sequencing to our testcase did not result in meaningful results and was not helpful to derive a product development plan. In contrast, the new augmented DSM sequencing algorithm delivers nearly the same process as the experienced domain experts. This seems to be a suitable approach for a future computer-based "intelligent engineering development planning".

The proposed methodology could be extended and used to estimate the processing times for each task and each process. Then, from the generated proposals for a processing order, a time dependent product development plan could be generated. If, in addition to the work tasks, milestones are included in the process-based DSM and suitably linked via relationships with the existing elements, the semi-automatic generation of milestone plans is also conceivable.

Most important, this promising new DSM method should be applied to other products, preferably by other researchers, before a general conclusion can be drawn. This applies especially for the selection and evaluation of different dependency types.

Acknowledgment

We like to thank CONTECS engineering services GmbH for the support of this research, especially for the data of the engine model. We are particularly thankful to Robert Lallinger, for his detailed insights about engine development and dimensioning.

References

Austin, S.A. (2000), “Analytical Design Planning Technique (ADePT): a dependency structure matrix tool to schedule the building design process”, *Construction Management and Economics*, Vol. 18 No. 2, pp. 173-182. https://doi.org/10.1080/014461900370807

Bartolomei, J., Cokus, M., Dahlgren, J., de Neufville, R., Maldonado, D. and Wilds, J. (2007), *Analysis and Applications of Design Structure Matrix, Domain Mapping Matrix, and Engineering System Matrix Frameworks*. [online] Massachusetts Institute of Technology. Available at: http://ardent.mit.edu/real_options/Real_opts_papers/Jennifer%20mini%20thesis.pdf

Browning, T. (2001), “Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions”, *IEEE Transactions on Engineering Management*, Vol. 48 No. 3, pp. 292-306, https://doi.org/10.1109/17.946528

Danilovic, M and Browning, T (2007), “Managing complex product development projects with design structure matrices and domain mapping matrices”, *International Journal of Project Management*, Vol. 25, pp. 300-314. https://doi.org/10.1016/j.ijproman.2006.11.003

Eppinger, S. and Browning, T. (2012), *Design Structure Matrix Methods and Applications*, Springer, Berlin.

Gebala, D. and Eppinger, S. (1991), "Methods for Analyzing Design Procedures", *Proceedings of the Third International ASME Design Theory and Methodology Conference, Miami, USA, September, 1991*, pp. 227-233.

Gopsill, J., Snider, C., Emanuel, L., Joel-Edgar, S. and Hicks, B. (2017), "Automatic design structure matrices: A comparison of two formula student projects", *Proceedings of the 21st International Conference of Engineering Design (ICED17) Vol. 6: Design Information and Knowledge, Vancouver, Canada, August 21 – 25, 2017*, The Design Society, Glasgow, pp. 31–40, https://doi.org/10.15125/BATH-00351

Gutierrez Fernandez, C.I. (1998), Integration analysis of product architecture to support effective team co-location, Master's Thesis, Massachusetts Institute of Technology.

Hildebrand, S. (2015), Umsetzung und Anpassung der DSM-Algorithmen für Clustering und Sequencing sowie der MDM-Funktionalität in ein Java-Programm, Bachelor's Thesis, Technische Universität Berlin.

Kreimeyer, M. and Lindemann, U. (2011), *Complexity Metrics in Engineering Design*, Springer, Berlin. https://doi.org/10.1007/978-3-642-20963-5

Lindemann, U. (2016a), *Handbuch Produktentwicklung*, Carl Hanser Verlag GmbH & Co., München. https://doi.org/10.3139/9783446445819

Lindemann, U. (2016b), *Sequencing a DSM*. [online] Technische Universität München. Available at: http://www.dsmweb.org/en/understand-dsm/technical-dsm-tutorial/partitioning.html (accessed 08.12.2017).

Maurer, M. (2007), "Structural Awareness in Complex Product Design - The Multiple-Domain Matrix", *Proceedings of the DSM Conference, DSM'07 / 9th International Design Structure Matrix Conference, Munich, Germany, October 16–18, 2007*, Institute of Product Development, TU München, München

Pahl, G. and Beitz, W. (2013), *Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*, Springer, Berlin. https://doi.org/10.1007/978-3-662-09186-9

Pimmler, T.U. and Eppinger, S.D. (1994), "Integration analysis of product decompositions", *Proceedings of the 6th International Conference on Design Theory and Methodology (ASME), Minnesota, USA, September 11 – 14, 1994*, The Design Society, Glasgow, pp. 343-351.

Schellert, D.D. (2014), Methodische Unterstützung zur Bestimmung einer optimalen Modulumsetzungsreihenfolge einer vorgegebenen Produktroadmap, Bachelor's Thesis, Technische Universität Berlin.

Steward, D. (1981), "The Design Structure System: A Method for Managing the Design of Complex Systems", *IEEE Transactions on Engineering Management*, Vol. EM-28 No. 3, pp. 71-74. https://doi.org/10.1109/TEM.1981.6448589

Thebeau, R.E (2001), Knowledge Management of System Interfaces and Interactions for Product Development Processes, Master's Thesis, Massachusetts Institute of Technology.

Thumm, B. and Göhlich, D. (2015), "Product Architecture Design Methodology for Developing Standardized Modules", *Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 7: Product Modularisation, Product Architecture, systems Engineering, Product Service Systems, Milan, Italy, July 27–30, 2015*, The Design Society, Glasgow.

Thumm, B., Göhlich, D., Orth, C., Presser, N. and Grammel, S. (2014), "Product Architecture Development and Modularization Process for Brownfield Development", *Proceedings of the ASME 2014 International Mechanical Engineering Congress and Exposition IMECE2014 Volume 11: Systems, Design and Complexity, Montreal, Quebec, Canada, November 14-20, 2014*, ASME. https://doi.org/10.1115/IMECE2014-37862

Yassine, A. (1999), "Engineering design management: an information structure approach", *International Journal of Production Research*, Vol. 37 No. 13, pp. 2957-2975. https://doi.org/10.1080/002075499190374

Yassine, A. (2004), "An introduction to modeling and analyzing complex product development processes using the design structure matrix (DSM) method", *Urbana*, Vol. 51 No. 9, pp. 1-17.

Prof. Dr.-Ing. Dietmar Göhlich, University Professor
Technische Universität Berlin, Methods of Product Development and Mechatronics
Strasse des 17. Juni 135, 10623 Berlin, Germany
Email: dietmar.goehlich@tu-berlin.de