INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

ALIGNING MULTIPLE DOMAINS OF DESIGN PROCESSES

M. Kreimeyer[1], M. Eichinger[1] and U. Lindemann[1]
[1]Technische Universitaet Muenchen, Institute for Product Development, Germany

ABSTRACT
Concurrent engineering processes are, in reality, a network of interlinked elements belonging to different domains such as process steps, information objects, organizational units, IT-resources, milestones, durations, decisions, and more. When trying to understand such a complex system, e.g. in business process reengineering projects, common approaches often regard only a single domain and therefore neglect other interdependencies that often turn out to be just as crucial. Using examples from a current project with a major German automotive manufacturer, an approach using design structure matrices as well as domain mapping matrices (combining them to obtain a Multiple Domain Matrix (MDM)) is shown to represent the existing multitude of process elements in a common model. The multiple domains can then be reduced to a single-domain view, which allows further examination of indirect process-structures (e.g. misalignment between indirect and real organizational structures). Examples for possible types of misalignment are given by comparing the as-is situation with the obtained results from MDM analysis in a case study.

Keywords: Design Structure Matrix, Domain Mapping Matrix, design process, dependency

1 Introduction
Process improvement projects are a common phenomenon in industry. Periodical redesign of internal procedures and organization are typical for all kinds of industry, sometimes as part of continuous improvement strategies, sometimes as part of process business reengineering projects [9]. While goals in process reengineering vary according to a company's strategy, a typical focus is to streamline processes to reduce lead-time, improve quality and employ only the resources necessary [13].

In the past, a lot of research has been carried out to develop process models and modeling tools to represent these various aspects (lead time, media continuity, optimum sequence of tasks,…). Examples are well-known methods such as SADT, IDEF, EPC, DSM or others [16, 10, 21, 2]. These are supported by a multitude of modeling tools, for example Staffware, SAP Netweaver, ARIS Toolset, and more. Furthermore, a lot of work has been done researching how to improve design processes, i.e. how to use the models designed in any of the methodologies lined out above to methodically improve a company's procedures when designing a technical product.

This paper presents a matrix-based approach to analyze existing models of the design process, enabling the user to compare existing direct and indirect linkages to each other in order to trace matching and misaligned process structures. For this purpose, dependencies within and across different domains of a design process model are extracted from an exemplary EPC model and modeled using a Multiple Domain Matrix [17]. From this, the matching of direct and indirect linkages is computed. The overall approach is further illustrated on a case study from automotive body design.

2 Concurrent Engineering
Concurrent design processes are considered state of the art nowadays, being applied to shorten the development time, to reduce the number of iterations and to foster collaboration ensuring a high degree of quality as well as early certainty about concepts that concern multiple disciplines ("design for X") [20].

2.1 The networked nature of concurrent engineering

A typical concurrent engineering process contains several different kinds of classes of entities, referred to as “domains” for the purpose of this paper [3]. Each domain regroups elements of similar kind (see [24] for the closer explanation of the concept of domains). Possible domains are e.g.:

- business objects (i.e. artifacts)
- tasks
- organizational units (e.g. departments)
- resources (e.g. IT-systems)
- milestones / decision points

Above all, the increasing maturity of the product to be designed is represented by a number of business objects or artifacts, such as data files, product models and other pieces of information that originate from the design process [19]. As a sum, these objects contain all information about the product. They are generated by a series of tasks, processing the artifacts as input and output [6]. Being supported by a number of resources such as test rigs, computer soft- and hardware, engineers and other personnel interact with the business objects through tasks [21]. The progress of time cannot only be recognized in the increasing maturity of the product representation but also in meeting milestones or when making decisions. Depending on the granularity of the model and the aspects incorporated, a number of domains as these can be found.

Looking into typical models as the EPC model [21] that concentrate on balanced modeling of processes, several typical domains of process modeling and improvement become apparent, in particular process steps or functions, the business objects exchanged during the process, the resources used, and the departments or people involved. These four are usually taken as the crucial ones and will be followed-up on in this paper.

Yet, concurrent engineering is not simply made concurrent by the simultaneous execution of tasks but by sharing business objects, by establishing organizational structures that support communication across department frontiers and by making resources compatible among each other where necessary. This means that there is not simply one flow of information whose processing is parallelized as much as possible. As each domain is built as a network of its own, concurrent engineering forms a network across different domains, which are, each one of them, a network of entities in themselves, too.

These networks are not independent of each other. In fact, the dependencies within each domain can support or counteract the dependencies within another domain. Well-arranged team structures that match the dependencies of the product structure, for example, can cater for a better outcome of the process [22, 11].

It is the basic hypothesis of this research that the dependencies between the different domains of a process need to be aligned in order to achieve a smoothly running process as suggested by [22]. As shown in the following, this means that, for example, team (or organizational) structures need to match the flow of information within the process.

2.2 Process modeling for concurrent engineering

There are a number of methods suitable for analysis and optimization of single aspects of a process, e.g. PERT or GANTT [12, 8]. Depending on the kind of process optimization project, a variety of aspects can be modeled (lead time, media continuity, optimum sequence of tasks,…). Typically, during larger endeavors, existing processes are modeled using process modeling techniques such as SADT, IDEF, EPC, DSM, or others [16, 10, 21, 2]. Yet, these models often only allow for single aspects to be modeled, and they are only helpful if certain aspects are sought for – as pointed out. Especially in the case of larger process maps, a lot of experience is necessary not only to model but especially to interpret the knowledge inherent to these charts. See figure 1 for an example: Without actually knowing what is modeled (i.e. having created the model oneself), it is virtually impossible to find a starting point for process improvement.

Models like the House of Business Engineering (HOBE) [21] suggest an overall, holistic approach to processes but do not allow for a balanced methodical analysis and optimization. However, this is necessary to understand the interconnections between the different domains in question in concurrent engineering and to harmonize the different domains.

In turn, a methodical approach is necessary for the systematic assessment of processes to help spotting weak points and to support finding room for improvement. Especially in the case of large process charts, manual analysis and optimization bears little fruit.

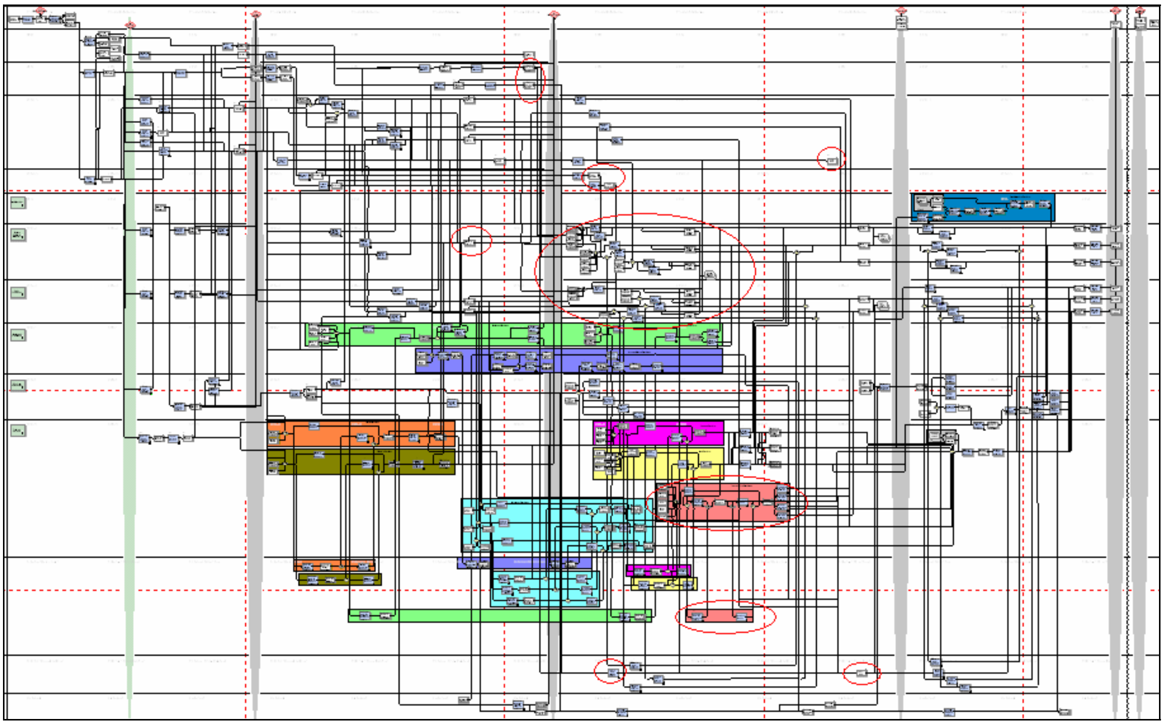

Figure 1: design process modeled in EPC (example to show overall complexity)

3 Multiple Domain Matrices for process modeling

Complexity management is practiced in many facets. One of them is the use of Design Structure Matrices (DSMs) to interrelate entities of one single domain among each other [23], [2]. They are supplemented by Domain Mapping Matrices (DMMs) that interrelate two domains at a time [4]. Together, they form Multiple Domain Matrices (MDMs) that can be understood as a system's description involving several perspectives onto the system simultaneously [7, 17].

	business object	process step	IT-system	organizational unit
business object		DMM		
process step	DMM		DMM	DMM
IT-system		DMM	DSM	
organizational unit		DMM		DSM

Figure 2: MDM notation (example)

Process models such as the EPC model shown in figure 1 only display part of the dependencies, feigning a more linear character than they actually bear. A typical process chart as the one shown follows the dependencies between tasks and business objects in an alternating pace. However, e.g. the IT-systems used for different tasks linked by interfaces, too, which enables the exchange of business objects even across the boundaries of direct flows of information as represented by the dependencies shown in the chart.

At the same time, models like these are little accessible for algorithmic interaction. Of course, the databases underneath the visible layer are accessible for algorithmic manipulation for the skilled user. For the purpose of understanding and analyzing the networked structure, however, the MDM method is much more suitable. Models as the EPC model above can easily be transferred into an MDM

notation without loss of information. This is, in fact, true for a large number of modeling techniques, e.g. IDEF-3, EPC, SADT and others that are similar in information content [8]. Figure 2 shows a possible setup to interrelate the elements incorporated in the EPC model shown. The MDM notation is not only more suitable for the application of algorithms; it also shows all dependencies simultaneously, thus better visualizing the networked character of the design process.
Usually, process models detailing the flow of information in a process incorporate decision points. To involve these (typically represented as Boolean operators in a process model) in such a matrix, several strategies can be followed. For a global assessment of the process network, decision points are of less importance and can be ignored, whereas for the analysis of local features, e.g. how one decision impacts feedback loops, they cannot be ignored. It is also possible to resolve each decision into a new separate matrix, as proposed by [1]. Table 1 groups possible strategies.

Table 1: strategies for involving decision points

operators as separate entity	**separate process matrix for each decision**	**ignoring decisions**
+ course of process (“behavior“) can be followed	+ simple representation in consistent model + course of process and structural features represented	+ structural features easily accessible + consistent representation
- not consistent - algorithms difficult to apply	- large amount of data (many matrices) - only purposeful for small models	- ignores dynamic features (“behavior”)
purposeful for highly dynamic models and in case of highly nondeterministic behaviour of process	purposeful for small model involving few decision points, especially if decision points are of high relevance	purposeful for large models and if decisions only impact part of the overall process network

From the new matrix, sub-graphs can be extracted to visualize partial networks. Touchgraph-like visualizations have proven most useful as an intuitive user interface for complex structures [18]. Figure 3 shows how an EPC model is transferred into an MDM. From that, a graph involving the flow of artifacts and the process steps van be derived. Current research has only looked into representing one domain as a graph at a time. This form of representation, however, is not focus of this paper.

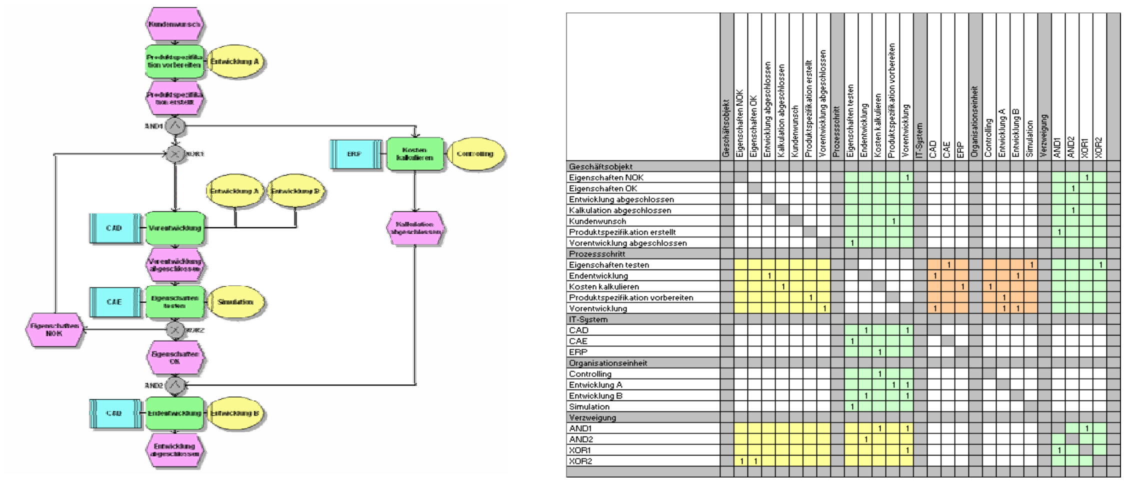

Figure 3: Exemplary design process (left) and equivalent MDM-representation (right)

4 MISALIGNMENT OF MULTIPLE DOMAINS OF A DESIGN PROCESS

A major advantage of modeling the process the way shown in section 3 is that indirect interdependencies between different elements of the process, in particular across different domains, become easily accessible. These play a major role, as typically two elements of the same domain (e.g. two information objects) are only linked in a different domain (e.g. because they appear in the same process step or are generated by the same person).

4.1 Critical objects

The concept that entities within the different domains of a process are to be harmonized or “aligned” was proposed before by [22]. With the scope of aligning structures of two or more different domains, those objects that are not in line have to be found. These are critical objects. They are determined by having a number of critical relations, i.e. they are not directly connected to other objects in their own domain but they are linked via objects existing within another domain. Figure 4 shows the schematics of why document 1 and document 2 are critical objects: Although no direct link relates the two, they are indirectly linked via process step 1. In the example, documents 1 and 2 belong to the domain of business objects, whereas the process step could be part of a domain named “tasks”.

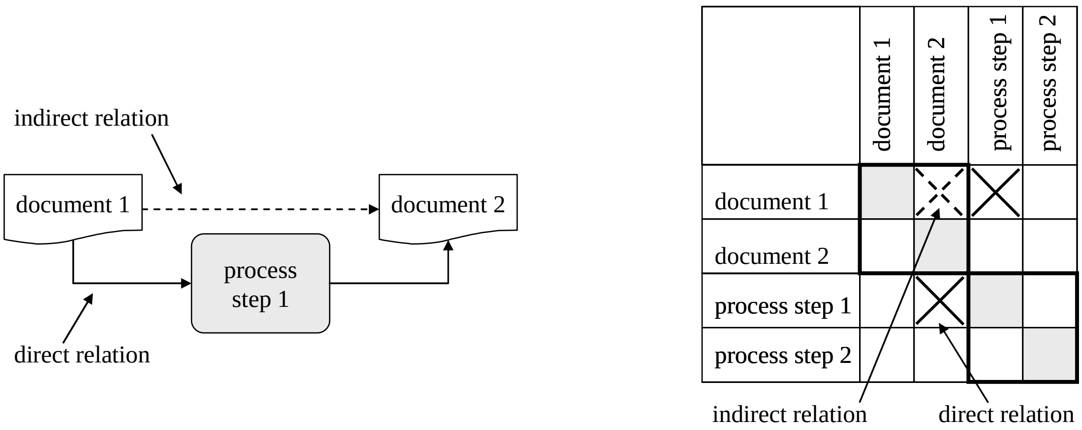

Figure 4: Example of indirect relation between two documents (flow chart and MDM)

Using an algorithm proposed by [7], the number of intra-domain dependencies due to cross-domain dependencies can be depicted directly. The more intensely and indirectly elements are interlinked, the more critical they are to the process; in case of the simple example in figure 4, both documents turn out to be critical, as they both have indirect links (to each other).
Therefore, the method allows for the detection of critical process elements. This alone provides a more detailed insight into how the overall process is structured, as it allows determining how intensely a change to an entity impacts onto other elements.

4.2 Identifying misaligned objects

In a second step, it is possible to compare the explicit structure in one domain (e.g. team relations modeled in a team based DSM [2]) to the computed indirect dependencies. This can point to where prevailing structures need to be questioned or extended by those that only exist indirectly.
The example in figure 5 shows how the organizational structure, broken down into a hierarchy of four sub-departments, is set up. The upper left matrix represents this hierarchy (notation: “row has link to column”). Equally, the sequence of three process steps is represented. Each process step is executed by one or two organizational units, as shown on the lower right part of the figure. Thus, it can be computed how the organizational units are related via process steps. In case of process step 3, organizational unit 2 is the only one executing this step. In process step 2, organizational units 1 and 4 are involved, having already a direct link as shown in the matrix. Ultimately, organizational units 2 and 4 collaborate on process step 1; however, this is not represented in the organizational chart, as they are only linked via organizational units 1 and 0. Hence, an indirect relation (shown as a dashed line) can be detected that is not matching the prevailing structure. This is a case of misalignment.
In general, misaligned structures therefore allow for the detection of dependencies that ought to be in place but are not. In the example presented in figure 5, this could mean that the link between organizational units 2 and 4 should be closer investigated. A common team structure, the establishment of a community of practice or the regular exchange of both units in a meeting might be possible consequences. Again, these collaborations could be represented in a matrix-based notation to

closer investigate the nature of that one indirect relation or to follow-up on how the misalignment is dealt with. In fact, many research projects investigating how collaboration between different fractions of a company can be rendered more efficient come to establishing this kind of link. Another example is presented in the following section.

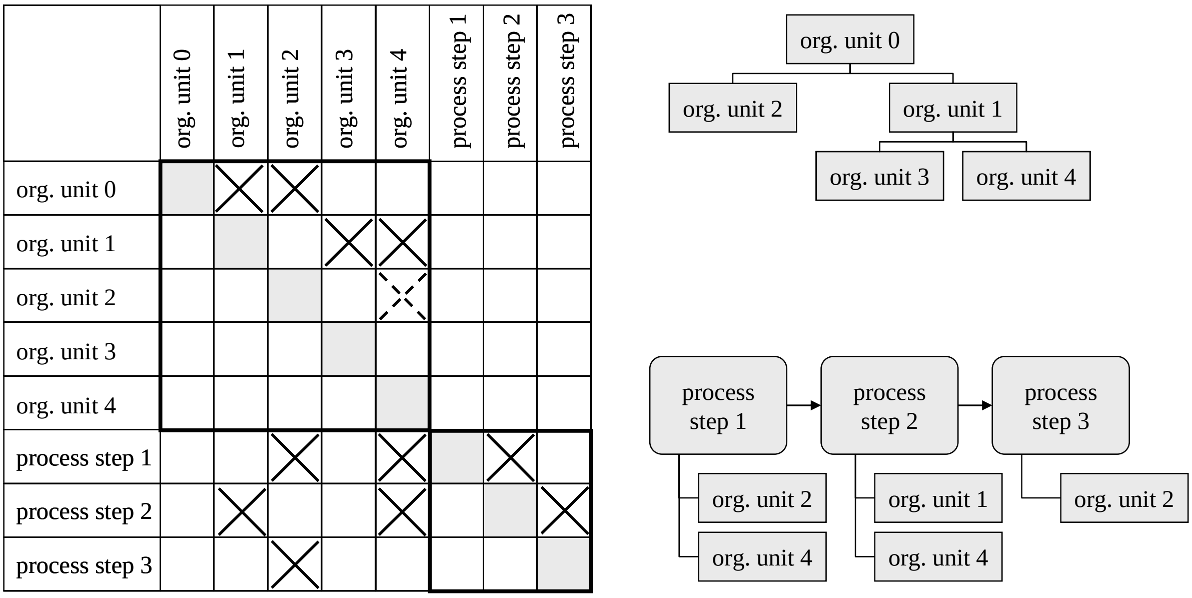

Figure 5: Comparing explicit structures in one domain to indirect ones

5 CASE STUDY

To validate the approach taken, a case study was carried out using a large process model representing the design process of major German automotive manufacturer. In detail, the collaboration between a number of departments concerned with the embodiment design and simulation of the car's body was analyzed. The scope of the process is the concept and serial development of a premium class SUV.

As an initial part of a research project investigating the nature of this collaboration, a process chart was created to represent the tasks, business objects, departments and IT-systems used. It also included further objects that are not regarded in this case study. The overall chart is shown in figure 1. The project focusing on the research on collaboration between embodiment design and simulation is explained in detail in [5].

The process model was made concentrating particularly on collaboration across the borders of the different departments. Hence, all involved entities are densely interrelated and business objects are exchanged continuously to be worked on by different departments, using different IT-systems. The overall model contains 49 process steps, 58 business objects, 20 organizational units (=departments), and 18 IT-systems. EPC is an object oriented process modeling method, therefore many of these objects within the model are instanced a number of times, especially business objects. In fact, the original model was reduced for computational reasons. Originally, it included 155 process steps and 128 different business objects.

Applying the approach explained above, a large number of critical objects showing misalignments on multiple levels can be detected. Figure 6 shows the MDM matrix representing these. The first (pink) domain contains business objects, the second (yellow) domain regroups process steps, the third (green) one comprises IT-systems and the last (blue) domain includes the different departments.

From this matrix, it can easily be seen that a large number of business objects are critical (about 80%). In fact, only those objects that act as an interface to further process steps not included in this model are not critical as such. While this may seem trivial, the matrix shows, in fact, that there are few objects that are more critical than others (shown dark red), as they are connected via several indirect linkages. These are, in fact, the core business objects that characterize the very essence of collaboration between the departments regarded. They represent the core artifacts that result from the process in focus. Supportingly, when analyzing the process for feedback loops as part of sequencing [2], using an algorithm by [15], most of the elements established as being critical also turn out to be involved in the majority of feedback loops of the flow of information, which backs up their importance. Using the dependencies of business objects as detected, the impact of changing one object onto the overall process can easily be seen. Each time, the cause of dependencies between business objects can be traced back to common process steps.

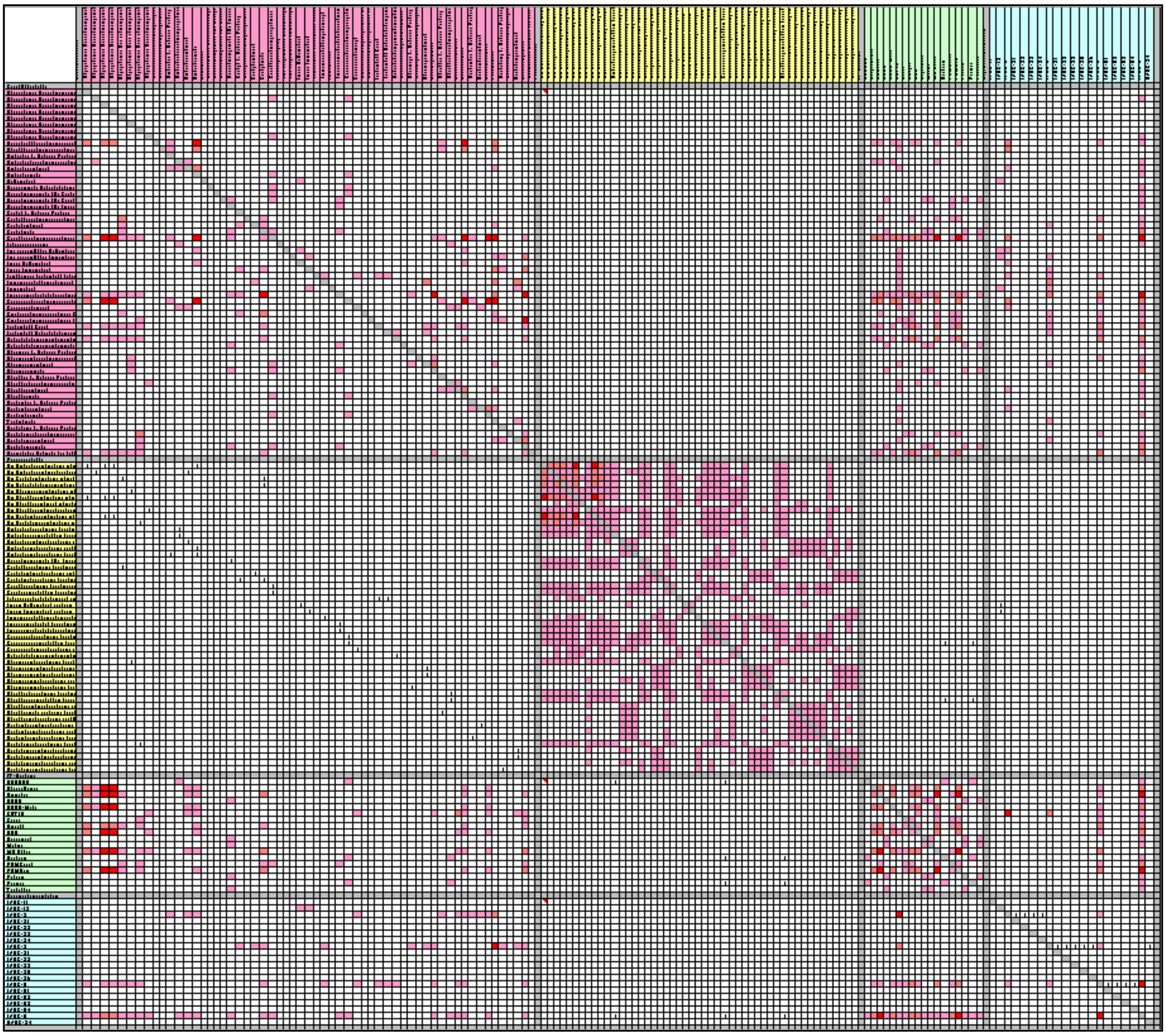

Figure 6: MDM with computed indirect relations (general overview to represent the number of indirect relations found in a design process)

The same analysis can, for example, be done for the organizational structure, being regrouped in the lower right corner of the MDM in figure 6. Figure 7 shows the matrix in more detail. Only few misaligned critical relations come up. There are three critical relations (they are not directed, so they cannot be named "dependencies"; hence the matrix is symmetrical) among

- organizational units 3 and 14,
- organizational units 8 and 14, and
- in particular between organizational units 14 and 20.

For nondisclosure reasons, the original denotation has been changed. Organizational unit 3 is the department responsible for the sheet metal design of the body-in-white, i.e. the car's body structure including doors, hatches, and vitrification. Organizational unit 8 develops all interior casings, including seats, dashboard, and trunk liner. Organizational unit 14 designs automotive safety features, i.e. active (e.g. airbags) and passive (e.g. crashworthiness) body features. Ultimately, organizational unit 19 is concerned with all FEM-based simulations (air flow, rigidity and safety of passengers).
Of the three indirect relations detected, one has, while this research was done implemented during a major reorganization project. During this project, the simulation department (organizational unit 19)

was disintegrated and its members were spread out – according to the nature of their tasks – over the other three departments. Those simulation engineers specialized on safety features are now grouped with organizational unit 14, those specialized on internal or external properties of the car are part of organizational units 3 and 8. At the same time, the close indirect relation of organizational units 3 and 8 to unit 14 is currently being in focus as to how collaboration can be improved, as their responsibilities overlap. In fact, this reorganization thus validates the findings shown in the matrix in figure 7, as the consequence of the three detected indirect linkages would be exactly what has been achieved by carrying out the reorganization project.

	Org.-Unit 1	Org.-Unit 2	Org.-Unit 3	Org.-Unit 4	Org.-Unit 5	Org.-Unit 6	Org.-Unit 7	Org.-Unit 8	Org.-Unit 9	Org.-Unit 10	Org.-Unit 11	Org.-Unit 12	Org.-Unit 13	Org.-Unit 14	Org.-Unit 15	Org.-Unit 16	Org.-Unit 17	Org.-Unit 18	Org.-Unit 19	Org.-Unit 20
Org.-Unit 1																				
Org.-Unit 2																				
Org.-Unit 3				1	1	1	1													
Org.-Unit 4			2																	
Org.-Unit 5			2																	
Org.-Unit 6			2																	
Org.-Unit 7			2																	
Org.-Unit 8									1	1	1	1	1							1
Org.-Unit 9								2												
Org.-Unit 10								2												
Org.-Unit 11								2												
Org.-Unit 12								2												
Org.-Unit 13								2												
Org.-Unit 14															1	1	1	1		
Org.-Unit 15														2						
Org.-Unit 16														2						
Org.-Unit 17														2						
Org.-Unit 18														2						
Org.-Unit 19																				
Org.-Unit 20								2												

Figure 7: Indirect linkages (pink and red) in the organizational structure

6 CONCLUSION

The method shown is, as was pointed out, suitable to detect misalignments of structures that are supposed to be harmonized and therefore makes the dependencies between the different domains of simultaneous engineering accessible. It furthermore is able to suggest necessary new interrelations as a starting point for a possible implementation of an optimized scenario.

6.1 Summary

Using the method proposed in this paper, multiple levels of interdependencies within concurrent engineering can be assessed at a time. By turning the indirect relations in a process into explicit ones, it turns one’s attention to possible critical objects and their interdependencies. In comparison to existing relations, the misalignment between the as-is situation and indirectly caused dependencies can be identified easily to discover not harmonized process elements. The method therefore complements existing procedures to describe and analyze processes.

Establishing the networked structure of concurrent engineering design processes and how they are modeled, a method was explained to represent these processes in the form of a Multi Domain Matrix to better visualize the networked character of these processes as well as make them better accessible for algorithmic interaction. Based upon this matrix, critical elements were identified, being those entities of the process (e.g. business objects or process steps) that are related (in a directed or non-directed fashion) indirectly via at least one entity of a different domain. Ultimately, these newly discovered indirect relations (being made explicit) were compared to the actually existing structure present in the process to uncover possible misalignments. The mode of operation was shown using a case study from automotive body design to present misalignments that later were the basis for restructuring a number of departments.

6.4 Reflection

Using the method originally proposed by [7] and applied here to look deeper into concurrent engineering design processes, multiple levels of interdependencies within concurrent engineering can

be accessed. Thus, the networked character of concurrent engineering can be better understood and controlled, as awareness of which entities are interlinked to what extent is raised. As it turns out, implicit interrelations play an important role in how people collaborate.
Globally, the modeling of several or even all domains of process allows for more holistic understanding. Hence, it contributes to a system's thinking in engineering management focused on design processes, complementing "classic" analysis methods (e.g. partitioning, cycles,…).

Implications for research

As misalignment between the actual situation and indirectly caused linkages can be identified, the method gives analytical access to the flow of information and its hindrances throughout the domains regarded. It therefore opens new perspectives for research on engineering design management, as it can draw links to how one domain (e.g. the team structure) should be configured to be in line with the other domains (e.g. the necessities to exchange information or business objects). However, while research looks into the cause and nature of things, the method presented rather sees its symptoms (e.g. what information exchanges are necessary) and what implications one should draw from it.
Stressing the web of relations between process entities, it also points out the true nature of concurrency. While concurrent engineering, or "simultaneous engineering", as it sometimes is referred to, is often understood as a purely temporal phenomenon, the fact that linkages also exist in non-temporal manners (such as IT-systems and their interfaces) stresses that concurrency actually rather means "networked".

Implications for engineering management

As stated in the previous section, the method shown raises awareness to how concurrent engineering is made concurrent by pointing the engineer's attention to possible critical linkages that impact a number of domains of a process simultaneously. When applied in process optimization projects, the method therefore can provide better understanding of how e.g. one work package or one process step is linked to a number of other elements, thus making the responsible personnel aware of how changes and conclusions from their work impacts other stakeholders in the process. For complex products, design engineers often find it difficult to judge the long range repercussions of their work (as e.g. addressed by the ForFlow project [14]).
Another possible side-effect is a check for coherence within the process model, as especially in larger models single relations or elements can be forgot easily. These can be easily detected using the algorithm to question the existence of (forgot) linkages.

Further research

Further research is currently done into the integration of a better visualization of cross-domain dependencies for more intuitive understanding and the consideration of dynamic aspects and decision logics. Also, the different kinds of implications for each domain need to be scrutinized further to gain better understanding of how concurrency actually impacts them. Further case studies can provide the necessary empirical data for this.

REFERENCES

[1] Belhe U, Kusiak A. Modeling Relationships Among Design Activities. *Journal of mechanical design*. 1996, 118, 454-460.

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

[3] Carlson Skalak S. *Implementing Concurrent Engineering in Small Companies*. 2002 (CRC Press, New York).

[4] Danilovic M. and Browing T. A formal approach for Domain Mapping Matrices (DMM) to complement Design Structure Matrices (DSM). In *The Sixth Design Structure Matrix (DSM) International Workshop*, September 12th – 14th , 2004, Trinity Hall College, University of Cambridge (Cambridge Engineering Design Centre, Cambridge).

[5] Deubzer F, Kreimeyer M, Herfeld U and Lindemann U. A Systematic Approach for Efficient Collaboration in Virtual Product Development Environments. In *16th International Conference on Engineering Design*, 28-31 August 2007, Paper 1007 (currently being published).

[6] DIN EN ISO 8402. *Quality Management*. 1995 (Beuth, Berlin).

[7] Eichinger M, Maurer M, Pulm U, Lindemann U. Extending Design Structure Matrices and Domain Mapping Matrices by Multiple Design Structure Matrices. In *8th Biennial ASME Conference on Engineering Systems Design and Analysis (ESDA)*, 4.-7. July 2006, Torino, Italy. 2006 (ASME, Torino).

[8] Fahrwinkel U. *Methode zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des business process reengineering*. 1995 (HDI, Paderborn).

[9] Havey M. *Essential business process modeling*. 2005 (O'Reilly, Beijing).

[10] IEEE Std 1320.1-1998. *IEEE Standard for Functional Modeling Language—Syntax and Semantics for IDEF0.* 1998 (IEEE, New York).

[11] Kreimeyer M., Deubzer F., Herfeld U., Dequidt C. and Lindemann U. Function-driven product design in virtual teams through methodical structuring of requirements and components. In *8th Biennial ASME ESDA 2006, Torino, Italy*, July 4-7, 2006 (ASME, Torino).

[12] Kreimeyer M, Ponn J and Lindemann U. Analysis and Benchmark of Methods for Process Modeling. In *9th International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology" TMT 2005*, 2005 (TMT, Antalya).

[13] Kubeck L. *Techniques for business process redesign - tying it all together*. 1995 (Wiley, New York).

[14] Lauer W, Ponn J and Lindemann U. Purposeful integration of product models into the product development process. In *16th International Conference on Engineering Design*, 28-31 August 2007, Paper 542 (currently being published).

[15] Lindemann U, Maurer M and Kreimeyer M. Intelligent Strategies for Structuring Products. *Engineering Design - Theory and Practice*. 2005 (Engineering, Design Center, Cambridge), 106-115.

[16] Marca D and MacGowan C. *SADT - structured analysis and design technique*.1989 (McGraw-Hill, New York).

[17] Maurer M and Lindemann U. Facing multi-domain complexity in product development. *CiDaD Working Paper Series*, 2007, 3(1). (http://www.pe.mw.tum.de/index.php?inhalt=cidad-wps)

[18] Maurer M, Pulm U and Lindemann U. Utilization of graph constellations for the development of customizable product spectra. In *Fourth International ICSC Symposium on Engineering of Intelligent Systems EIS 2004* Madeira (Portugal), 29. February - 02. March 2004. 2004 (ICSC Interdisciplinary Research, Madeira).

[19] Ponn J, Lindemann U. Characterization of Design Situations and Processes and a Process Module Set for Product Development. In 15th International Conference on Engineering Design – ICED 2005. Melbourne, 15.-18. August 2005 (Institution of Engineers Australia, Barton).

[20] Sapuan S M, Osman M R, Nukman Y. State of the art of the concurrent engineering technique in the automotive industry. *Journal of Engineering Design*, 2006, 17(2), 143-157.

[21] Scheer, A-W. *ARIS - Business Process Modeling*. 2000 (Springer, Berlin).

[22] Sosa M., Eppinger S. and Rowles C. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science*, 2004, 50(12), 1674–1689.

[23] Steward D. V. The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transactions on Engineering Management*, 1981, 28(3).

[24] Danilovic M. and Browning T. Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management*, 2007, 25, 300–314.

Contact: Matthias Kreimeyer
Technische Universitaet Muenchen
Insitute for Product Development
Boltzmannstrasse 15
85748 Garching
Germany
Phone: +49 89 28915136
Fax: +49 89 28915144
e-mail: matthias.kreimeyer@pe.mw.tum.de