10TH INTERNATIONAL DESIGN STRUCTURE MATRIX CONFERENCE, DSM'08
11 – 12 NOVEMBER 2008, STOCKHOLM, SWEDEN

MULTIPLE-DOMAIN MATRIX AS A FRAMEWORK FOR SYSTEMATIC PROCESS ANALYSIS

Carsten König[1], Matthias Kreimeyer[2] and Thomas Braun[3]

[1]Daimler AG
[2]Institute for Product Development, Technische Universität München
[3]Teseon GmbH

Keywords: MDM, Multi-Domain Matrix, business process, process management, process modeling

1 INTRODUCTION

Process management is an important success factor in any company nowadays, mostly introduced to reduce cost, to improve the time to market, and to raise quality, all while making the company more efficient in its procedures and more agile towards the customer. The goal of process management is to establish knowledge of one's own processes, to improve processes, to document them (e.g. for compliance reasons), to calculate the cost of running a process, and to define roles and responsibilities as well as interfaces between parts of a process. To do so, the planning and modeling of a process are core activities. Based on these models, processes are run, controlled, and analyzed to be improved.

There are numerous approaches available to support process management, and many different process models have come up to depict various aspects of process management. This paper looks into how a process can be modeled to be analyzed using a Multiple-Domain Matrix.

2 COMMON PROCESS MODELS

Typically, processes are modeled as flow charts that produce large process maps to describe how a company is progressing from a customer request to a finished product. To name only a few, e.g. SADT (Structured Analysis and Design Technique), IDEF (Integrated Definition), UML-Activity diagrams, BPMN (Business Process Modeling Notation), XPDL (XML Process Definition Language), PMM (Process Module Methodology [1]) or EPC (Event-driven Process Chains, either event-driven or object-oriented (oEPK) [2]) have become typical standards to model such processes, as well as PERT or GANTT representations. All of them focus on the flow of information through a series of activities. Table 1 provides an overview of some of these methodologies, regrouping the core semantic elements that, of course, vary slightly in their definitions but focus on basically the same domains within a process. As can be seen, almost all methodologies capture a process in terms of the interaction of tasks, documents, events, roles / resources, and time. Of course, the selection of domains is case-specific, and domains can be split or detailed hierarchically to include further information.

Table 1: Comparison of common process modeling methodologies

	Task				Document				Event					Role / Ressource					Time		
Process Modeling Methodologies \ Constructs	Function	Operation	Activity	Transition	Data Object	Attribute	Input	Output	Event	Message	Object	State	Places	Organizational Unit	Ressource	Attribute	Methods & Tools	Resources	Milestone	Lead Time	Start / End Time
UML	x		x		x	x	x	x	x		x			x	x	x	x		x		
EPC	x				x				x					x	x				x		
oEPK		x				x				x	x			x		x					
IDEF			x				x	x				x									
Petri-Net			x	x									x								
PMM			x				x	x									x	x	x		
PERT			x																	x	x

3 COMPREHENSIVE PROCESS ANALYSIS

Understanding the behavior of a modeled process under various perspectives can be crucial to understand the causal connections within highly interrelated processes. While all process models shown above start from the basic sequence of tasks, they focus on different perspectives on a process. To be able to do so, the models are, in fact, designed for a variety of different applications and foci.

3.1 Processes as multi-layered networks

All process models consist of (mostly directed) graphs that represent the relations between the entities of each of the domains shown in table 1. In fact, each domain represents a network in itself, e.g. documents have relations among each other, IT-systems have interfaces, organizational units are related mutually, and so on.

Each individual network is, in fact, part of the focus taken by a specific modeling methodology. However, the co-existence of these methodologies also shows that processes are multi-layered networks that consist not simply of a flow of information or a series of tasks. Rather, they consist of numerous domains that are networked internally and among each other. Only together, they form a complete process. The more comprehensive modeling methodologies, such as EPC or UML, provide for these different networks and relationship types, in a more or less integrated manner. However, even these lack systematic analysis support, which is available for matrix-based methodology.

To model a process comprehensively and to gain a deeper understanding of it, it should therefore be understood as the multi-layered network it actually is, i.e. it should comprehend all the company organization that actually is necessary to enable it. To analyze a process, in turn, it is important to select and relate all domains that are relevant to such a specific analysis in an integrated manner that, at the same time, enables and facilitates systematic and comprehensive analysis.

4.2 MDM-based Process Analysis

Multiple-Domain Matrices (MDM) [3] allow for representing multiple network structures, both within a single domain (e.g. tasks) and across domains (e.g. roles: what resources a task relies on). Equally, MDM is able to capture different relationship types that coexist simultaneously. This way, a process that has multiple networks within different domains or that is using different relationship types within one domain can easily be represented.

An MDM process-model can be assembled from a number of sources. This way, different process models can be combined, or an existing model can be completed from other sources; e.g. the organizational setup can be attached to the tasks, or different process models, even if modeled in different modeling methodologies, can be assembled into an overall model.

Using MDM as a modeling scheme to represent a process gives a number of advantages:

- The network structure of the process is modeled in all its facets. This way, no single relationship dominates over the others, i.e. the complexity of the process is captured more realistically. In fact, most process models can be converted into a MDM without much or with no loss of information. Of course, the model should only be built to the extent necessary, respecting the common lessons learned available for process models [4].
- Different models can be combined; this way, it is possible to check how well-aligned the different structures that are modeled actually are [5] (e.g. comparing process to team structures).
- Qualitative and quantitative models can be combined to some extent, if they each can be represented using MDM methodology (e.g. using weights for nodes or edges or by introducing attributes via additional MDMs).
- The process can be analyzed either based on the native data (e.g. the DSMs and MDMs that the MDM is assembled from) or with a regard to the impact of an analysis onto other domains (e.g. by finding clusters in a task-task-DSM and then constituting teams in the role-role-DSM, as it could be done for the example in the slides that are part of this publication), or it can be analyzed using computed DSMs that incorporate indirect dependencies (e.g. by computing how tasks are interrelated via documents that serve as input and output, and then further analyzing the new task-task-DSM). In the latter way, a process can be analyzed more holistically by also accounting for dependencies across modeled entities that extend over more than one domain.
- Structural characteristics become accessible. This allows for the systematic analysis using all kinds of available algorithms for DSM, DMM and MDM analysis. Based on native or computed matrices, these structural characteristics can, again, span one or more domains.
- The process can be evaluated using complexity metrics. Using structural characteristics and different metrics based thereon [6], not only quantitative process models can be analyzed numerically, but also the mere existence of entities and their relations can be assessed and used to find possible weak spots.

- Different processes can be compared at an abstract level. Using either complexity metrics or looking at the occurring structural characteristics, different processes can be put side by side.

Of course, MDM-modeling brings a number of downsides, too:

- The matrices grow very rapidly. While theoretically almost all information in a common process model can be converted into a MDM, this makes hardly sense. If e.g. an EPC model contains many attributes, e.g. starting- and end-times of every function, already a small process chart will turn into a very large MDM if all attributes are transferred. Rather, it makes sense to only convert those parts of a process model that are of interest to process analysis.
- The actual readability of matrices is very limited. Purely modeling a process in MDM notation therefore rarely makes sense, as most users will be unable to understand the process model. Thus, little transparency would be generated.
- The actual graphical structure of a flow chart is lost when turned into a MDM. This is a major shortcoming, as the "structuredness" and "style" of the flow chart layout are important for the understandability of a process model [7], as they transmit part of its meaning, too.

4 CONCLUSION

As shown, MDM can serve as a powerful tool for process modeling and analysis, extending common rather flow-oriented approaches to an understanding that views a process as a multi-layered network of various domains. While it helps to overcome noticeably the problems of analyzing a process more holistically and in-depth using the existing means of computational DSM, DMM and MDM analysis, it brings along large and complex models that are impossible to read and understand without IT support. The tradeoff as to what is preferable and necessary depends on each individual project context and should be made with regard to common modeling standards [4].

Further work needs to look into how interaction with such models can be made more intuitive, above all, to extend MDM from a tool that is oriented to the analysis of structures to a methodology that embraces all aspects of managing a complex process structure.

REFERENCES

[1] Bichlmaier C. and Grunwald S. PMM - Process Module Methodology for Integrated Design and Assembly planning, In *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, 12.-16.09.99.

[2] Scheer A.-W., Nüttgens M., Zimmermann V. Objektorientierte Ereignisgesteuerte Prozeßkette (oEPK) Methode und Anwendung. In Veröffentlichungen des Instituts für Wirtschafts-informatik, Heft 141, 1997 (Saarbrücken, Instituts für Wirtschaftsinformatik).

[3] Maurer, M. *Structural Awareness in Complex Product Design*, 2007 (München, Technische Universität München).

[4] Schütte R. *Grundsätze ordnungsmäßiger Referenzmodellierung. Konstruktion konfigurations- und anpassungsorientierter Modelle*, 1998 (Wiesbaden, Gabler).

[5] 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.

[6] Kreimeyer M., König C., Braun T. Structural Metrics to Assess Processes, In *10th International DSM Conference* 2008, Stockholm, Sweden.

[7] Gruhn V. and Laue R. What business process modelers can learn from programmers. *Science of Computer Programming*, 2007, 65, 4–13.

Contact: Matthias Kreimeyer
Technische Universität München, Institute of Product Development
Boltzmannstrasse 15
85748 Garching
Germany
Phone +49 89 28915136
Fax +49 89 28915144
e-mail matthias.kreimeyer@pe.mw.tum.de

10TH INTERNATIONAL DSM CONFERENCE

Multiple-Domain Matrix as a Framework for Systematic Process Analysis

Carsten König[1], Matthias Kreimeyer[2], Thomas Braun[3]

[1]Daimler AG
[2]Institute of Product Development
[3]Teseon GmbH

Technische Universität München

JÖNKÖPING INTERNATIONAL BUSINESS SCHOOL

MANAGE COMPLEX SYSTEMS
FOLLOW THE FLOW OF INFORMATION!

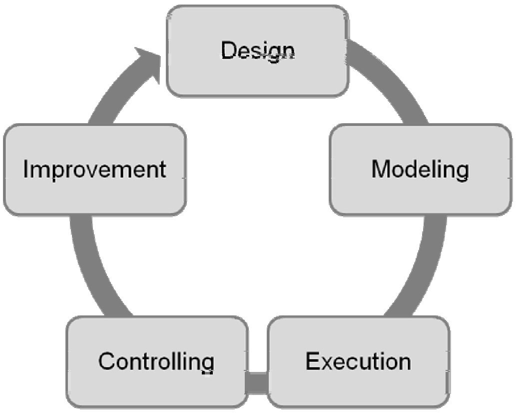
Index

Technische Universität München

JÖNKÖPING INTERNATIONAL BUSINESS SCHOOL

Introduction

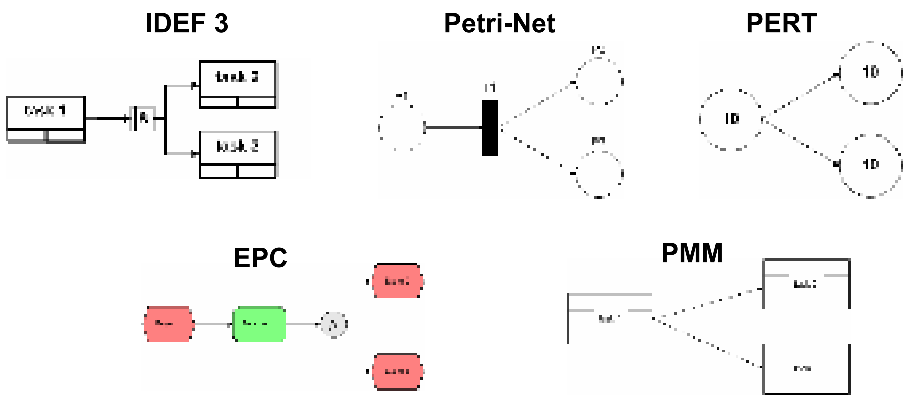
- Process modeling is the basis for any management of a process
  - Knowledge of one's own processes
  - Improvement of processes
  - Documentation
  - Calculation of cost
  - Definition of roles / responsibilities
  - Set-up of interfaces
- Life-cycle of process management
  - Designing a process
  - Modeling a process
  - Execution of a process
  - Controlling a process
  - Improvement of a process



FOLLOW THE FLOW OF INFORMATION!
MANAGE COMPLEX SYSTEMS

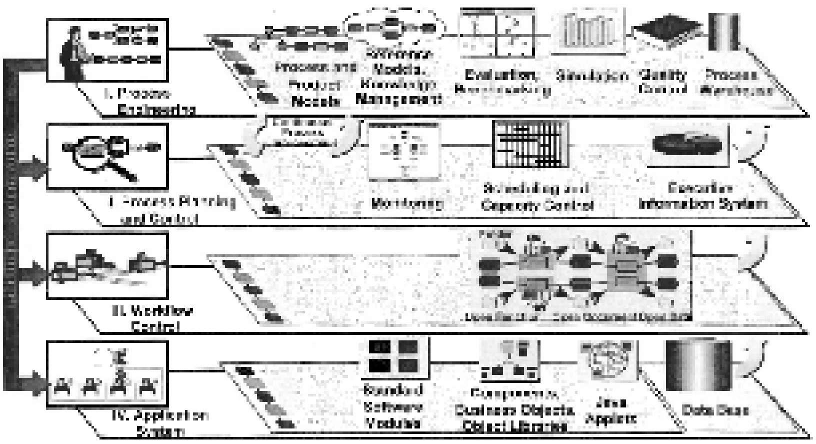
Common Process Models

- Common process models are mostly flow oriented
  - Sequence of tasks
  - Sequence of events
  - …



Common domains in process modeling

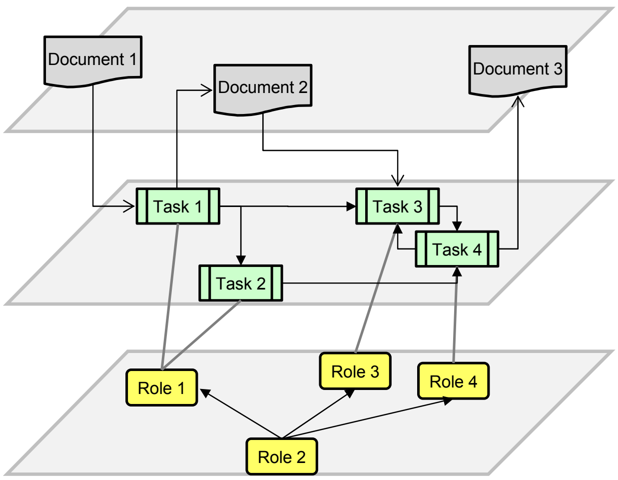
- Many perspectives on a process persist
  - Tasks, Information Objects, Events / States, Resources,…
- Comprehensive approaches to process modeling include these perspectives
  - Views on process represent different domains
  - e.g. ARIS House of Business Engineering (HOBE – see figure below)



Scheer, A.-W. ARIS – Business Process Frameworks 1998, Fig. 24

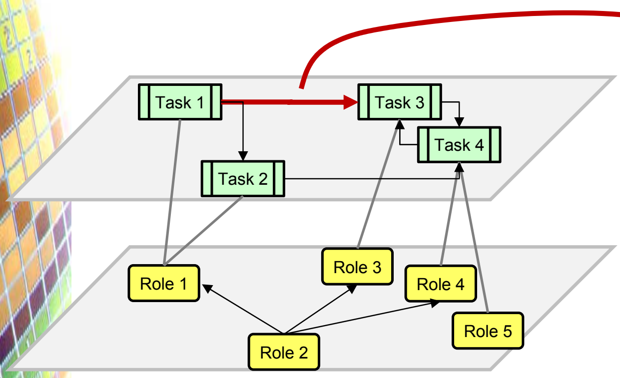
Processes as a network of multiple domains

- Entities in each domain of a process are commonly related to each other and to entities in other domains
- Core focus is classically on tasks and documents
- Models to represent networks in each domain exist



Processes modeling using MDM – model setup

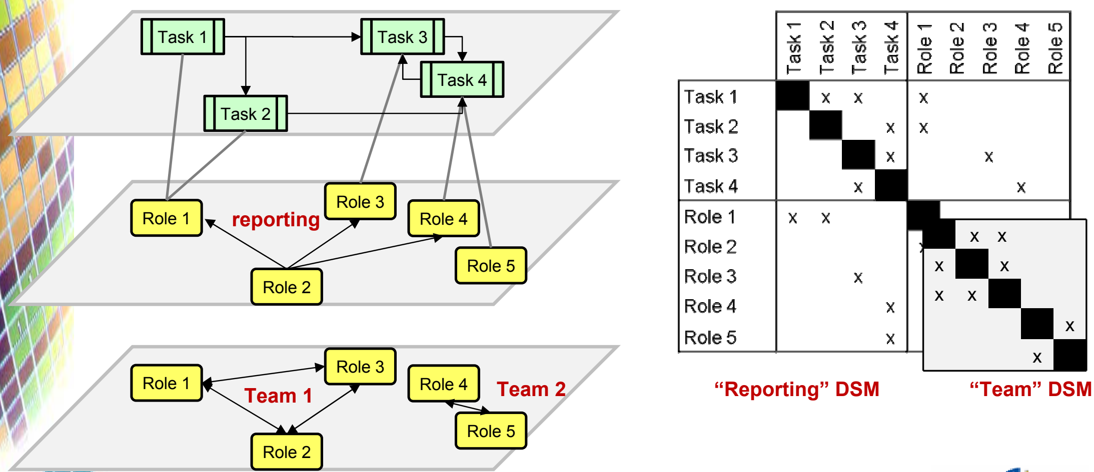
- Each domain can be represented as a DSM
- Each relationship across any two domains can be modeled as a DMM
- Combination of matrices into a MDM



	Task 1	Task 2	Task 3	Task 4	Role 1	Role 2	Role 3	Role 4	Role 5
Task 1	■	x	x		x				
Task 2		■		x	x				
Task 3			■	x			x		
Task 4			x	■				x	
Role 1	x	x			■				
Role 2					x	■	x	x	
Role 3			x				■		
Role 4				x				■	
Role 5				x					■

Processes modeling using MDM – relationship types

- Integration of various co-existing relationship types

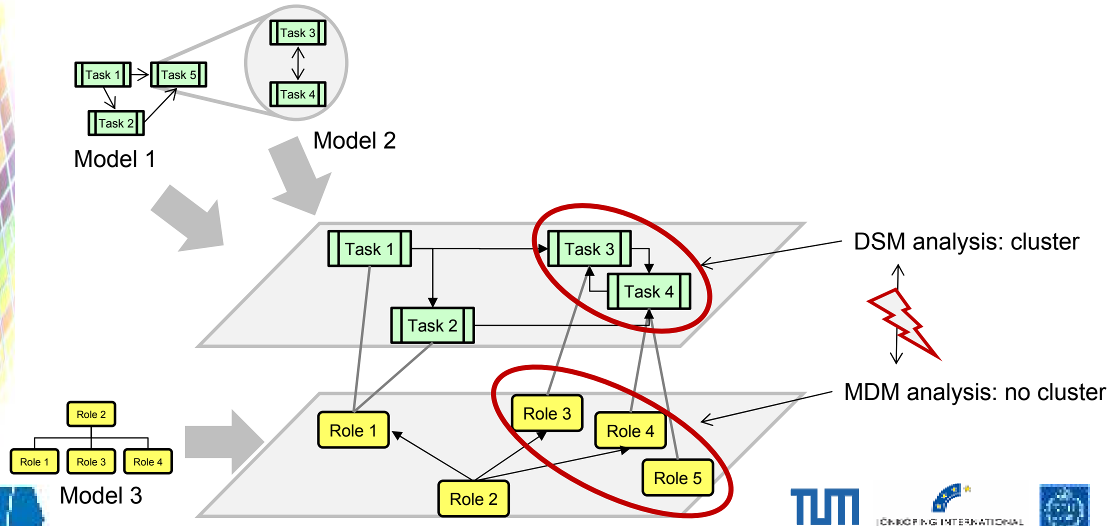


FOLLOW THE FLOW OF INFORMATION!

MANAGE COMPLEX SYSTEMS

Processes modeling using MDM – Chances

- Assembly of different process models
- Structural characteristics in DSM and DMM, e.g. clusters
- Patterns across MDMs, e.g. indirect dependencies
- Application of complexity metrics



Technische Universität München

10th International DSM Conference 2008- 9

FOLLOW THE FLOW OF INFORMATION!

MANAGE COMPLEX SYSTEMS

Example 1: Process Analysis in Engineering Release Management (1/4)

Goal: Improvement of communication

- Alignment of committees and organizational entities with process chain is needed
- Improvement of the assignment of tasks (“functions”) to organizational entities
  - i.e. re-shuffling the work distribution based on the process sequence
  - Deduction of interdependencies between organizational entities
  - Generation of suggestions for an improved organizational setup (Organizational Units)
- Improvement of the composition of committees
  - i.e. re-shuffling the committees based on the process sequence
  - Deduction of communication channels between committees

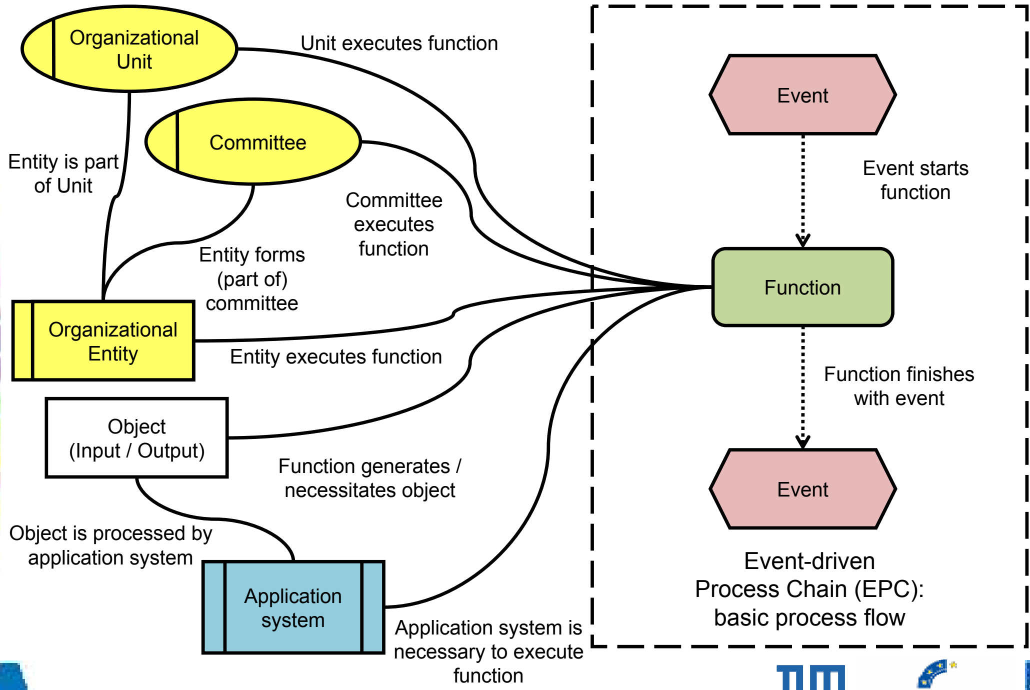
Process Model and Analysis

- New organization to be designed for new product line (automotive manufacturer)
- Transformation of EPC model for basic process into MDM
- Computation of indirect dependencies between organizational entities and committees to improve communication and minimize use of resources
- Improvement of data flow (documents and related IT systems)

Technische Universität München

10th International DSM Conference 2008- 10

Example 1: Process Analysis in Engineering Release Management (2/4)



Example 1: Process Analysis in Engineering Release Management (3/4)

	F	E	OU	C	OE	I/O	AS
Functions F		E takes place after F				F generates I/O	
Events E	E starts F						
Organizational Units OU	OU executes F						
Committees C	C executes F						
Organizational Entities OE	OU executes F		OE is part of OU	S forms C			
Objects I/O	I/O is necessary input for F						I/O is processed by AS
Application Systems AS	AS is necessary to execute F						

Example 1: Process Analysis in Engineering Release Management (4/4)

Analysis of the processes

- Analysis of computed organizational entities-organizational entities DSM (via functions) and computed committees-committees DSM (via functions)
- Analysis of computed objects-objects DSM (via functions)
- Detection of structural characteristics in computed DSMs (e.g. clusters, bottlenecks, start-/endnodes,…)
- Examination of fitness of native organizational structure compared to computed organizational entities–organizational entities DSM (alignment)
- Consideration of causes for indirect dependencies to derive improvements in organizational entities-functions DMM and committees-functions DMM ("why-matrix")

Findings

- Identification of potential for lean process chain:
  - Reduction of media breaks via information integration (integration in central application system)
  - Elimination of redundant communication efforts and definition of communication interfaces (OU delegates in committees)
- Quantification: Consideration of capacities of organizational entities

Example 2: Process Analysis in Product Design (1/4)

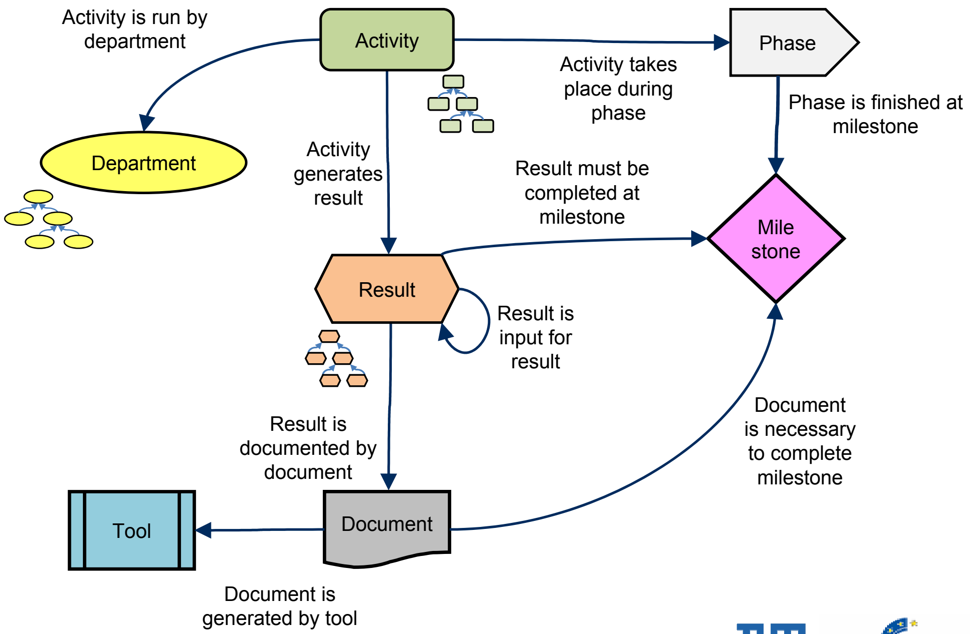
Goal: Cross-linking of activities in design process

- Interdependencies of different activities are unknown
- Alignment of tasks with milestones to reduce overruns of deadlines
- Reduction of iterations by improving the sequence of activities

Process Model and Analysis

- Design of a process guideline for a manufacturer of drive technology
- Basic process is documented in project handbook, additional information exists in checklists for milestones
- Substructures of three domains (activities, results, departments) available as hierarchical matrices
- Interlinking of activities through results obtained
- Configuration of process forerun for each milestone
  (i.e. ideal sequence and interdependencies of activities before each milestone)
- Detection of drivers of robustness of the process
  (i.e. elements that are critical to the success of the process)

Example 2: Process Analysis in Product Design (2/4)



FOLLOW THE FLOW OF INFORMATION!
MANAGE COMPLEX SYSTEMS

Example 3: Process Analysis in Product Design (3/4)

	A	R	D	T	OU	P	MS
Activities A		A produces R			A is run by OU	A takes place during P	
Results R		R is input for R	E is documente d in D				R must be completed at MS
Documents D				D is generated by T			D completes MS
Tools T							
Departments OU							
Phases P						Sequenceo f Ps	P is finished at MS
Milestones MS							Sequence of MSs

Example 2: Process Analysis in Product Design (4/4)

Analysis of the processes

- Analysis of acquired results-results DSM and computed activities-activities DSM (via results)
- Analysis of computed departments-departments DSM (via results)
- Detection of structural characteristics in native and computed DSMs (e.g. hierarchies, clusters, bottlenecks, start-/endnodes,…)

Findings

- Activities are mostly interlinked via results
- Identification of significant results as complexity drivers in results-results DSM: high number of hierarchies → elements with extensive impact chains
- Determination of critical documents
- Identification of potential for lean process chain: Improvement of assignment of responsibilities (department-department DSM is densly networked, because responsibilities are unclear and partially redundant)
- Installation of content-driven process forerun (evaluation of relevance of milestones depending on the results of a project)

Example 3: Process Assessment in Control Unit Design (1/4)

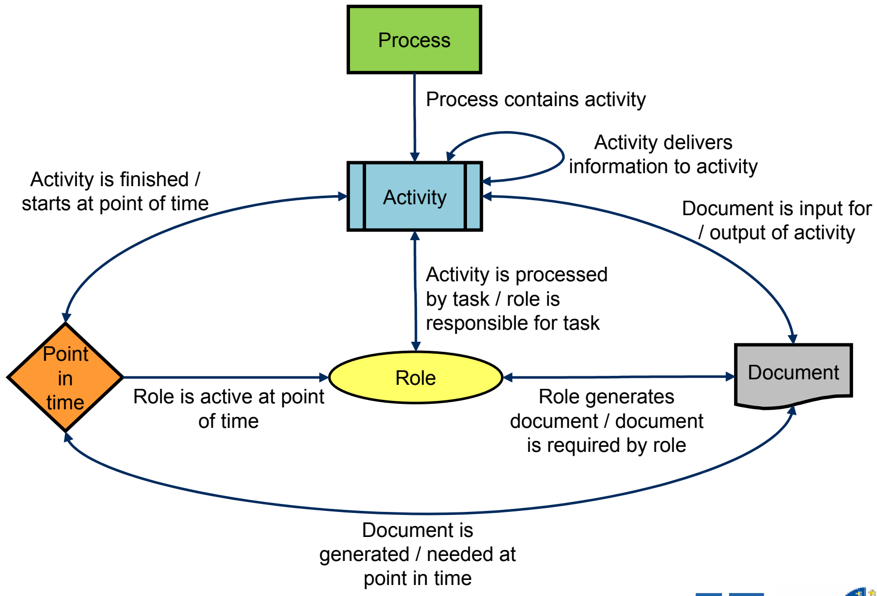
Goal: Analysis / comparison of 15 design processes forming one design process

- Weak spots in a very large design process consisting of 15 sub-processes unknown
- Determination of the structural quality of the processes using structural metrics
- Analysis of different structural features for all elements of the process using ABC-analysis to spot the most critical elements
- Qualitative comparison of the different processes

Process Model and Analysis

- Design process for control unit design in automotive development
- Assembly of the process model from 198 different models: different process models (mostly Innovator), automated parsing of different other descriptions (mostly DOORS), separate models for 15 processes, assembled into one global MDM
- Analysis using 24 different structural characteristics and 20 different metrics across most domains, based on 18 calculated DSMs out of the native data

Example 3: Process Assessment in Control Unit Design (2/4)



FOLLOW THE FLOW OF INFORMATION!
MANAGE COMPLEX SYSTEMS

Example 3: Process Assessment in Control Unit Design (3/4)

	P	A	R	D	T
Processes P		P contains A			
Activites A			A is processed by R	A has D as output	A finishes at T
Roles R		R is responsible for A		R generates D	R is active at T
Documents D		D is input for A	D required by R		D is needed at T
Points in time T		A starts at T		D is generated at T	

Example 3: Process Assessment in Control Unit Design (4/4)

Analysis of the processes

- Comparison of native task-task DSM to computed task-task DSM (via documents)
- Analysis of task-DSM, document-DSM, point-in-time-DSM via various structural metrics (native and computed matrices)
- Detection of structural characteristics in native and computed DSMs (e.g. hierarchies, clusters, start-/endnodes,…)
- Comparison of processes (native and computed task-task DSM per process) among each other via metrics

Findings

- Numerous models could be combined
- Data on roles too incomplete to obtain valid results
- Determination of critical tasks, documents, and points in time for each process (e.g. most central tasks, documents with highest impact on overall process, point in time that synchronizes highest number of tasks,…)

Conclusion

- Process analysis is rarely just focused on the sequence
  - Boundary conditions are set by other domains in process organization
  - Impact of change of a process on other domains
  - Indirect dependencies exist across all domains involved
- MDM is advantageous for a more "holistic" approach to process management
  - Assembly of various models
  - Representation and analysis of overall network
  - Application of available algorithms and metrics for complexity management
- MDM process model is difficult to use
  - Matrices are large and little intuitive to read
  - No graphical modeling