A META MODEL OF THE INNOVATION PROCESS TO SUPPORT THE DECISION MAKING PROCESS USING STRUCTURAL COMPLEXITY MANAGEMENT

Sebastian Kortler[1], Udo Lindemann[1]
(1) Institute of Product Development, Technische Universität München, Germany

ABSTRACT

The innovation process is characterized by numerous interactions of numerous domains. Cyclic interdependencies intensify the pressure in terms of quality and schedule, causing shortened testing phases, frequent releases of new models, and thus hardly calculable risks. Structural Complexity management is established in order to avoid wrong decisions, instable processes and error-prone solutions. Therefore, Structural Complexity Management evaluates system's characteristics by analyzing system's underlying structures across multiple domains, condensing each single analysis into one big matrix that represents multiple domains at a time.

Identifying suitable perspectives, generating suitable models and using suitable analyze criteria are the challenges in this field.

In order to support the manufacturing of innovative products and thus the evaluation and interpretation of the system's underlying structure this paper proposes a meta model. The created model describes the author's perspective on entities arising during the innovations process and their interactions. The proposed model is used to simplify the decision making processes and to enable the management of cyclic interdependencies during the innovation process.

Keywords: structural complexity management, structural criteria, structural meanings, cycle management, meta model

1. INTRODUCTION-MOTIVATION

Manufacturing technical innovative products implies complex design processes as well as complex product architectures with manifold challenges caused by cyclic interdependencies. Those cyclic interdependencies intensify the pressure in terms of quality and schedule, causing shortened testing phases, frequent releases of new models, and thus hardly calculable risks. The whole innovation process is characterized by numerous interactions of numerous domains. Moreover, manifold artifacts, models and actors are involved. Complexity management is established in order to avoid wrong decisions, instable processes and error-prone solutions. Structural Complexity Management evaluates system's characteristics by analyzing system's underlying structures across multiple domains, condensing each single analysis into one big matrix that represents multiple domains at a time.

However, comparing and evaluating the criteria of a complex structure makes it necessary to interpret different structural criteria and then evaluate their impacts on the system. Identifying suitable perspectives, generating suitable models and using suitable analyze criteria are the challenges in this field.

In order to support the manufacturing of innovative products and thus the evaluation and interpretation of the system's underlying structure this paper proposes the process of deriving a meta model. The created model describes the author's perspective on entities arising during the innovations process and their interactions. The proposed model is used to simplify the decision making processes and to enable the management of cyclic interdependencies during the innovation process. Especially the presented model is used as a systematical basis for Structural Complexity Management.

The paper is structured as follows: After defining relevant terms in section 2, a short review of the current research in structural complexity management is presented in section 3. Section 4 describes which scientific methods were used to derive the meta model. Section 5 presents the meta model. Section 6 demonstrates the use of the model. Finally, the paper proposes a conclusion.

2. DEFINITIONS

2.1 System

A system is created by entities (elements) and their interdependencies (relationships) forming a system's structure. Such a structure possesses individual properties, which contribute to fulfill the system's purpose [1]. Systems are delimitated by a system border and connected to their surroundings by inputs and outputs. Changes of system's parts can be characterized by dynamical effects, which lead to a specific system's behavior. However, in this paper variations over time are not considered.

2.2 Domain

Domains represent the classification of elements, which create the system. Examples of domains are "components" or "documents".

2.3 Relationship type

The relationship type describes the meaning of a dependency. Different relationship types can even exist between the same elements and between the same domains [2]. Examples of relationship types are "change impact" or "waiting for".

2.4 Structure

"Structure" is understood as the network formed by dependencies (edges) between a system's entities (nodes). It furthermore relates to the semantics of this network; the structure of a system therefore always contributes – in its constellation – to the purpose of the system. Structures and their subsets can be analyzed by means of computational approaches, primarily provided by the graph theory and related sciences [2].

2.5 Structural criteria

A structural criterion is understood as a particular constellation of nodes and edges, i.e. it is formed by a particular pattern considering nodes and edges [2]. The criterion gains its meaning by the way the pattern is related to the actual system it is part of, i.e. it must serve a special purpose in the context of the overall system [1]. A structural criterion only possesses significance in the context of the system it is describing.

2.6 Structural meaning

Structural meanings relate structural criteria to their respective effects impacting the modeled system. The effects are, amongst other factors, dependent on the modeled domain, the relationship type describing the dependencies between the corresponding entities (see figure 1).

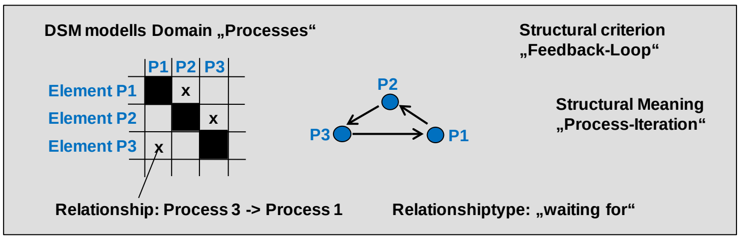

Figure 1. Definition of structural meanings

2.7 Cycle

Cycles are reoccurring patterns of temporal or structural nature that can be subdivided into phases. Their constituting elements are:

- Repetition
- Phases
- Duration
- Triggers
- Effects

3. RELATED WORK

3.1 Structural Complexity Management

To manage a structure efficiently, different methodologies prevail: Most commonly, matrix based methodologies such as the Design Structure Matrix (DSM), Domain Mapping Matrix (DMM), and Multiple Domain Matrices (MDM) are commonly applied, and the underlying theory provides for ample means of analysis. Furthermore, network theory is available, describing how the structure of random systems in nature, which have evolved over time, can be described. Ultimately, graph theory provides for a formal, mathematically founded framework grasping complex interdependencies.

Network and graph theory are closely interconnected. Hence, it is not easy to separate them. Whereas network theory focuses on the global features of any network, graph theory addresses structural features that originate from the interaction of single nodes and edges of a network structure. Graph theory is often traced back to Euler's works (e.g. [3]), while network theory can be dated back to the research of [4].

Research on matrix based complexity management has come a long way. Originating from a process focus with the first published formulation of a DSM [5], a whole community has developed around this research. The DSM is able to model and analyze dependencies of one single type within one single domain. Browning [6] classifies four types of DSMs to model different types of problems: component-, team, activity-, and parameter-based DSMs. However, many other classifications exist (e.g. [2]) nowadays.

There are numerous algorithms to analyze the overall structure of the relationships within a DSM; starting from the original algorithms for tearing, banding and partitioning [7][8] to a still non-exhaustive list provided by [2].

The authors of [9] have extended DSM to DMM, i.e. Domain Mapping Matrices. The goal was to enable matrix methodology to include not just one domain at a time but to allow for the mapping between two domains, as previously postulated e.g. by [10]. [2] has taken this approach further to model whole systems consisting of multiple domains, each having multiple elements, connected by various relationship types. He refers to this approach as Multiple Domain Matrix (MDM). He provides a number of ways to analyze the system's structure across multiple domains, condensing each single analysis into one DSM that represents multiple domains at a time. That way, he is able to apply algorithms for DSM analysis meaningfully across several domains, i.e. across a whole system. As especially the last DSM conferences have shown, matrix-based approaches integrating multiple views "domains" become more and more accepted to manage several perspectives onto a system, especially when it comes to large structures (e.g. >1000 elements per DSM).

3.2 Interpretation of structural criteria

Most of the approaches of structural complexity management look into what criteria qualities can be found in a structure, from the level of a global structure down to the integration of individual nodes. Structural criteria relates to the pattern of nodes and edges. Figure 2 orders the structural criteria, as provided by [2], by the evaluation of the number of edges and nodes that form a structure. In fact, most of the criteria can be traced back to a few basic elements [11][12].

In [2], several structural criteria are identified and interpreted considering change propagations between the elements regarding the modeled domain "components". Therefore, Maurer [2] divided structural criteria depicted in figure 2 into 2 groups: Structural criteria describing the meaning of nodes and edges and structural criteria describing the meaning of subsets. For each of these groups Maurer [2] discovered the structural criteria's meanings considering the development of a race car. The author presented how structural meanings ease structural complexity management by suggesting several interpretations of structural criteria. Until today several structural meanings considering different domains are identified:

Eben [13] analyzed structural meanings of requirements. Elezi [14] identified structural meanings considering processes with the aim of lean thinking. Kortler [15] described structural meanings considering components and their responsible designer. Kortler [16] identified another structural meaning considering the connection of requirements and design artifacts.

But, the structural meanings cannot be transferred from one application to another by implication. Different structural meanings may occur caused by differing data acquisition methods or differing models. In order to ensure a systematically basis a meta model describing which domains interact with each other considering the innovation process can be useful.

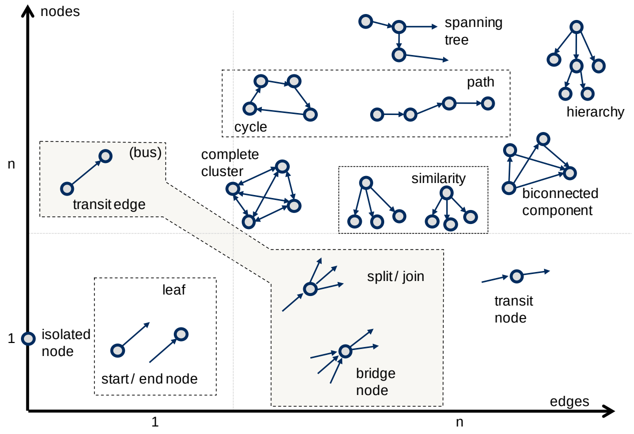

Figure 2. Basic structural criteria [11] [12]

4. SCIENTIFIC APPROACH

The authors are part of a research project which focuses in the area of managing cycles in innovation product development processes. The central topic of the research is the implementation and use of elements of complex solutions, nowadays typically consisting of a combination of product- and service-components, so-called product-service-systems (PSS). The components are subject to development-, manufacturing-, and life-cycles of varying length which are provided by different functional divisions. Availability and maturity of technologies, changes of competences, financial cycles at capital markets or of investments and write-offs as well as changes of customer demands represent external influences on the company. In contrast, the associated business processes underlie different cycles in research and development, manufacturing, logistics, finance, service, and recycling, which are mutually affecting each other as well.

The collaborative research project is performed by an interdisciplinary team from the fields of engineering, social and business sciences at TU München. The project is divided into different subprojects. Each of these subprojects focuses on different perspectives of the innovation process.

In order to derive a meta model describing the interactions between involved entities, the authors performed 14 interviews. Having performed the first interview phase the authors derived about 50 domains with diverse interactions. In order to remove the identified redundancies and gaps of the first version, the authors designed a catalog including gaps and redundancies in the meta model. With the help of this catalog the authors performed a second interview phase focusing the interactions of the modeled domains. Finally, 35 domains were identified. Considering the projects view, the elements of these domains are identified as the most influencing elements during the innovation process. Moreover, the authors derived more than 450 possible relationship types between elements of these domains. The second version of the model was used to implement the subproject’s views on cycles during the innovation project. Thereby, the authors derived the most important domains and dependencies included in innovation cycles of the respective subprojects. In conclusion the authors classified the identified domains and derived a group-model in order to understand the essential dependencies of the basic entities during the innovation process.

5. A META MODEL DESCRIBING THE PROJECTS VIEW OF THE INNOVATION PROCESS

Figure 3 depicts the domains whose elements and dependencies are used in the presented meta model (depicted in figure 5b). Table 1 illustrates all of the abbreviations used in the meta model. The meta model depicted in figure 5b represents the project’s view of the most important entities and interactions inside the innovation process. The development processes are not described in detail as a meta model describing development processes in detail is provided by Kreimeyer [12]. The proposed model is however not complete considering the innovation process. Instead of that, the first benefit of the model is supporting the author’s project by highlighting entities and dependencies which are involved in re-occurring events (so-called cycles) – the dependencies shown with orange color in figure 5b. The presented model collects the most relevant relationship types of the identified domains. Dependencies between elements of the same domain are not included in this model. With the presented model the authors can decide whether changes in one or more elements will lead to change propagations in other elements. Thus, the authors can use the model in order to support the decision making process (taking change propagations into account).

With the help of the presented model and the 'model of four aspects in product development' [2] the authors classified the identified domains. To in order do so, the authors enhanced the product aspect to the Product-Service-System (PSS) aspect. Figure 3 depicts the identified domains ordered to their groups. Moreover, the authors identified for each group its own time reference. Elements of the process domains act in different phases of the innovation process. Elements of the PSS domains can be ordered to one or more PSS life cycle phases. Elements of the group environment or organization-unit have their own lifecycle (machines become obsolete, employee leave the company, laws change over time).

The authors identified super-relationship types describing the basic dependencies between the four groups. Subsequently the authors combined the groups and relationship types to a meta² model (depicted in figure 4). This model describes how the elements of the domains of the groups interact with other elements in general. As a second benefit, the model leads the authors to an improved understanding of the whole innovation process.

As a third benefit, this gained knowledge can be enhanced with structural knowledge in order to derive structural meanings describing the impacts of cycles or the possibilities of managing cycles during the innovation process. Thus, the model can be used as a systematic basis combined with structural complexity management. In doing so the authors identified structural criteria and structural meanings regarding special dependencies between elements of different domains.

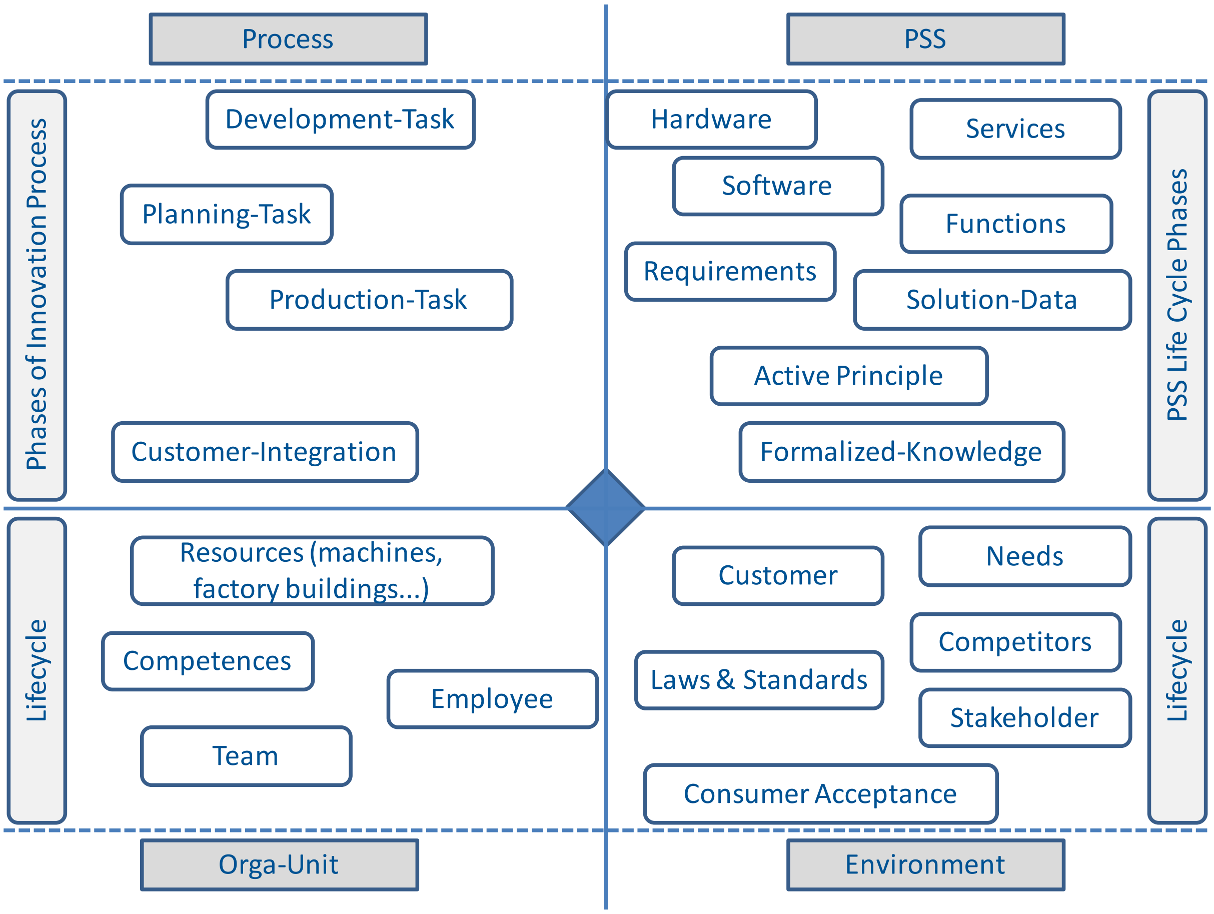

Figure 3. Clustered domains of the meta model

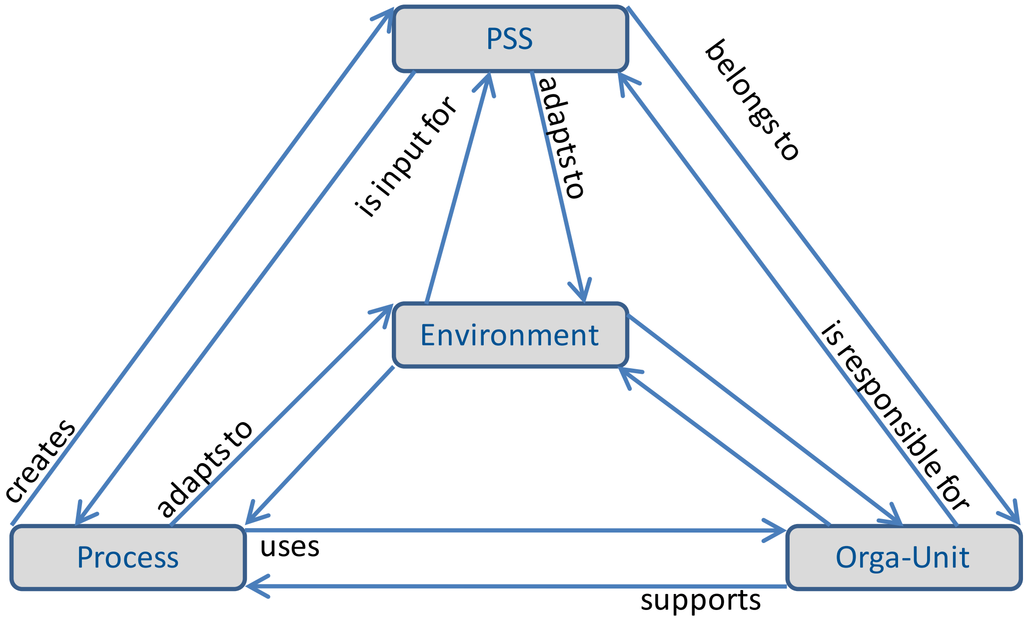

Figure 4. Meta² model

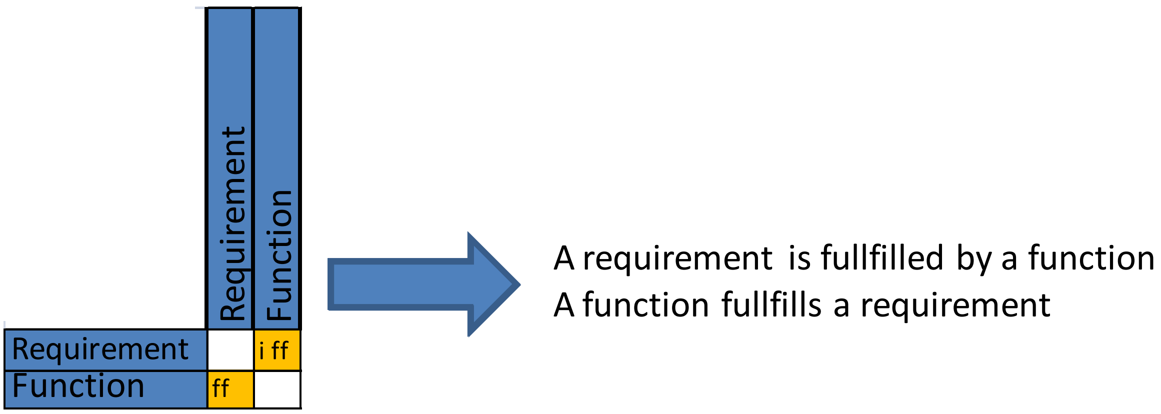

Figure 5a. Small Section of the meta model

	Requirement	Function	Service	SW-Component	HW-Component	Active Principle	Solution-Data	Variant	Requested Specification	Mandatory Specification	Goal	Meta-Model	Rule	Sequence of Rules	Constraint	Planing-Tasks	Development-Tasks	Production Technology Planning	Adaption of Production Structures	Design of Changeable Production Resources	Production and Assembly Task	Customer Integration Task	Production Resources	Component of Production Resource	Technology	Technology Chains	Production Structure	Competence	Employee	Team	Customer	Need	Customer's Accepting Factor	Competitor & Stakeholder	Law & Standard
Requirement		i ff	i ff	i ff	i ff	i ff		bel	i col		em	i dec	usd		ir	der	inp	der	der	der												ent		ent	ent
Function	ff		i ff	i ff	i ff	i ff	em	ipo		i col	ff	i dec	usd		ir		inp												i ds	i ds				ent	
Service	ff	ff	use	cp	cp		em	ipo		i col	ff	i dec	usd		ir		i ds												i ds	i ds		ent			
SW-Component	ff	ff	cp	use	cp		imp	ipo		i col	ff	i dec	usd		ir		i ds												i ds	i ds				i bo	
HW-Component	ff	ff	cp	cp	use	use	imp	ipo		i col	ff	i dec	usd		ir		i ds				iasma		i ma	i ma	c ma	c ma	c ma	ne	i ds	i ds				i bo	
Active Principle	ff	ff			i us		i us	i us		i col	ff	i dec	usd		ir		inp																		
Solution-Data		ent	ent	ent	ent	in		i us			ff						inp					i id													
Variant	ff	ip	ip	use	use	use	use	in		i des	ff	i dec	usd	der			i ds	inp	inp	inp	iasma		i ma	i ma	c ma	c ma	c ma	ne	i ds	i ds				ent	
Requested Spec.	in							des		i ff	use					i cr	inp												i ds	i ds	inp			inp	
Mandatory Spec.		in	in	in	in	in		des	ff								i ds												i ds	i ds	i pa			i pa	
Goal	ent	ent	i ff	i ff	i ff	i ff	use	i ff	ipo								inp																		
Meta-Model	def	def	def	def	def	def						use	pro b		def		i us																		
Rule	desu	desu	desu	desu	desu	desu						des i		ipo	desu		i us																		
Sequence of Rules													in	use			i us																		
Constraint	ri	ri	ri	ri	ri	ri						i def	i con				i us																		
Planing-Task								cr	cr		cr					i fo	tri	con	con	con	con	use						use	i per	i per		con	con	con	con
Development-Task	cr	cr	cr	cr	ds	uses	uses	des	an	cr	uses	cr	cr	cr	cr	i tr	i fo	con	con	con	con	use						use	i per	i per		con	con	con	con
Prod-Technology P.	set							i us								pro d	pro d		pro i	pro i	ds				con	gen		use						con	
Adapt Prod-Struct	set							i us								pro d	pro d	pro d		pro i	ds		an		an		gen	use						con	
Design of Prod. Res	set							i us								pro d	pro d	pro i	pro i		ds		ad	comb		use		use						con	
Prod & Assembly					asma			asma								i con	i con	i ds	i ds	i ds	i fo		use	use	use	use	use	use	i per	i per				con	con
Customer Integr							id									i us	i us					sup									int	id	id		
Prod. Resources	set				asma			asma											i con	i ad	i us			in	imp		ipo	ne	i us						ad
Comp of Prod. Res	set				asma			asma												i cr	i us		i co	comp	uses			ne							ad
Technology	set				asma			asma										i con	i con		i us		i us	i us	rep	ipo	ipo	ne							ad
Technology Chain	set				asma			asma										i gen		inp	i us		uses		in		ipo	ne							ad
Production Structure	set							con											i gen		i us		con		con		rep								ad
Competence																i us	i us	i us	i us	i us	i ne		i ne	i ne	i ne	i ne									
Employee	i res	i res	i res	i res	i res	i res		i res	i res	i res						per	per				per		i res	i res	i res	i res	i res	offers	wo	ipo					ad
Team	i res	i res	i res	i res	i res	i res		i res	i res	i res						per	per				per		i res	i res	i res	i res	i res	offers	in	co					
Customer									pro/re	pro/re												i in										off	owns	bel	
Need	inp		i ff													i con	i con					i id									i of				
Cust. Acc.Factor																i con	i con					i id									owned				
Competitor	set	set		pro	pro			pro	pro/re	pro/re						i con	i con	i con	i con	i con	i con										has				
Law & Standard	set															i con	i con				i con		r	r	r	r	r		r						

Figure 5b. The created meta model including identified domains and relationship types

Table 1. Table of abbreviations used in the meta model

Abbreviation	Relationship Type	Abbreviation	Relationship Type
ado	adapts	i ma	is manufactured by
an	analyzes	i ne	is needed
asma	assembles / manufactures	i of	is offered by
bel	belong to	i pa	is passed to
c ma	can be manufactured with	i per	is performed by
co	consist of	i res	is responsible for
comb	combines	i tr	is triggered by
comp	is compatible with	i us	is used for/by
con	considers	iasma	is assembled / is manufactured
cp	complements	id	identifies
cr	creates	imp	implements
def	defines	in	includes
der	derived by	inp	is input for
des	describes	int	interviews
des i	describe instantiation	ip	implements
desu	describes usage	ipo	is part of
ds	designs	ir	instantiation is restricted by
em	emerge from	ne	needs
ent	entails	nm	needs to be manufactured with
ff	fulfills	off	offers
gen	generates	per	performs
i ad	is adapted by	pro	provides
i bo	is bought at	pro b	Provides building blocks for
i co	is compatible with	pro d	provides data for
i col	is collected	pro i	provides input for
i con	is considered	r	restricts
i cr	ic created by	re	receives
i dec	is declared by	rep	replaced by
i def	is defined by	ri	restricts instantiation
i des	is described in	set	sets up
i ds	is designed by	sup	supports
i ff	is fulfilled by	tri	triggers
i fo	is followed by	use	uses
i gen	is generated by	usd	usage is described by
i id	is identified by	wo	works with
i in	is interviewed by		

6. USING THE META MODEL

The aim of the meta model is to support the development of innovative products and thus the evaluation and interpretation of the system’s underlying structure. Therefore, the meta model provides a systematical basis. The authors’ project aims on managing cycles which appear during the innovation process. At this point the authors present a small example using the presented meta model and structural complexity management in order to manage such emerging cycles.

This example demonstrates the applicability throughout the iterative process of refining requirements and concretizing product properties [16]. In this case the meta model prepares the dependencies between requirements and functions (see figure 5b).

The aim of the authors [16] was to control the refinement cycles of requirements and functions. Therefore, they connected stepwise the requirements model to the functional model. Finally they identified the structural criteria (active sum and passive sum) as an instrument to control the refinement cycles. More precisely, they indicated which function and which requirement need to be refined within which iteration. To do so, they mapped the requirements model on the functional model

by using inter domain matrices. All functions and requirements as well as their relations are captured within each step of iteration. In order to identify functions and requirements with a high potential of refinement, they used active and passive sum considering the inter domain matrices. In this way the possibility of controlling the refinement cycles in each step of iteration was provided. The presented meta model (depicted in figure 5b) was used as a systematical basis for Structural Complexity Management. The model in this paper highlights domains and relationship types where the interpretation of structural criteria would be useful. Thus, the interpretation of structural criteria can be supported by using the proposed model. Scientists can start identifying further structural meanings.
Moreover, the designed meta model is to generate more transparency in the way of acting for all influencing stakeholders inside the innovation process. With the help of the meta model, all the participants of the innovation process can easily derive whether performed actions will have any influences on other involved partners. Moreover, all involved parties can identify whether the change actions of other partners propagate on their own elements.

7. CONCLUSION AND OUTLOOK

Structural awareness becomes more important considering the development of innovative products. Small changes in structures of the innovation process can cause huge impacts, so all available information about structure should be used in order to avoid wrong decisions, instable processes and error-prone solutions. Structural Complexity Management assumes a systematical basis in order to derive stable and reusable structural interpretations. The created meta model represents such a basis. In future work the authors aim to identify important paths inside the meta model. These paths allow for deriving chains of effects. Furthermore, the authors will include the proposed time references of the four identified groups (depicted in figure 3) into the meta model. Another point is the systematical identification of deliverables being transferred between the groups. In future research the authors will be focused primarily on the interpretation of further structural criteria in order to ease the development of innovative products and to support cycle management.

ACKNOWLEDGEMENTS

We thank the Deutsche Forschungsgesellschaft (DFG) for funding this research as a part of the collaborative research centre “Managing cycles in innovation processes – Integrated development of product service systems based on technical products” (SFB 768).

REFERENCES

[1] Boardman, J. and Sauser, B. System of Systems - the meaning of of. System of Systems Engineering, 2006 IEEE/SMC.
[2] Maurer, M. Structural Awareness in Complex Product Design. Lehrstuhl für Produktentwicklung (Dr.-Hut, München, 2007).
[3] Gross, J.L. and Yellen, J. Graph Theory and its Applications. (Chapman & Hall/CRC, Boca Raton, 2005).
[4] Erdoes, P. and Rényi, A. On Random Graphs I. Publicationes Mathematicae Debrecen, 1959, 6 (290).
[5] Steward, D.V. The design structure system: A method for managing the design of complex systems. IEEE Transactions on Engineering Management, 1981, 28, 71–74.
[6] Browning, T. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. IEEE Transactions on Engineering Management, 2001, 48(3), 292-306.
[7] Kusiak, A. Engineering Design: Products, Processes and Systems. (Academic Press, San Diego, 1999).
[8] Steward, D. Partitioning and Tearing Systems of Equations. Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis, 1965, 2(2), 345-365.
[9] Danilovic, M. and Browning, T.R. Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of Project Management, 2007, 25(3), 300-314.
[10] Yassine, A., Whitney, D., Daleiden, S., Lavine, J., “Connectivity maps: modeling and analysing relationships in product development processes.” Journal of Engineering Design, 2003, 14(3), 377-394.

[11] Kortler, S., Kreimeyer, M. and Lindemann, U., “A Planarity-based Complexity Metric”, Proceedings of the International conference on engineering design, ICED 2009.
[12] Kreimeyer, M., “A Structural Measurement System for Engineering Design Processes” Lehrstuhl für Produktentwicklung (Dr.-Hut, München, 2010).
[13] Eben, K., Daniilidis, C. and Lindemann, U. “Interrelating and prioritizing requirements on multiple hierarchy levels”, Proceedings of 11th International Design Conference DESIGN 2010 Dubrovnik, Croatia 2010.
[14] Elezi, F. “Reducing waste in product development by use of Multi-Domain Matrix methodology”, Proceedings of 11th International Design Conference DESIGN 2010 Dubrovnik, Croatia 2010. Design Paper 2010
[15] Kortler, S., Diepold, KJ. and Lindemann, U., ”Structural Complexity Management using domain-spanning structural criteria”, Proceedings of 11th International Design Conference DESIGN 2010 Dubrovnik, Croatia 2010.
[16] Kortler, S., Helms, B., Berkovich, M., Lindemann, U., Shea, K., Leimeister, JM. and Krcmar, H., “Using MDM-Methods in order to improve managing of iterations in design processes”, Proceedings of the 12th International DSM Conference Cambridge, UK 2010

Contact: Sebastian Kortler
Technische Universitaet Muenchen
Institute of Product Development
Boltzmannstraße 15
D-85748 Garching
Germany
Phone: +49 89 289 151 53
Fax: +49 89 289 151 44
Email: kortler@pe.mw.tum.de
URL: http://www.pe.mw.tum.de

Sebastian Kortler is a scientific assistant at the Institute of Product Development at the Technische Universität München, Germany since 2008. Before, he studied computer science until 2007. Currently, his research is focused on structural complexity management and the development of structural characteristics that govern complex systems.

Udo Lindemann is a full professor at the Technische Universität München, Germany, and has been the head of the Institute of Product Development since 1995, having published several books and papers on engineering design. He is committed in multiple institutions, among others as Vice President of the Design Society and as an active member of the German Academy of Science and Engineering.