Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 16 (2013) 303 – 312

Procedia Computer Science

Conference on Systems Engineering Research (CSER'13)
Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

Systems Engineering Management Based on a Discipline-Spanning System Model

Jürgen Gausemeier[a]*, Tobias Gaukstern[a], Christian Tschirner[a]

[a]Heinz Nixdorf Institute, University of Paderborn, Fürstenallee 11, 33102 Paderborn, Gernamy

Abstract

In many current development projects targets concerning time, cost and quality are often not achieved. This is due to the complexity of the product and its engineering processes. The conceivable development of information and communication technology will enable advanced mechatronic systems. Their manifold system functions, the cross-linking of elements within the system and their hardly manageable interactions induce a much higher complexity in the development process and make it much more challenging than today. As approaches of Model Based Systems Engineering (MBSE) get more and more accepted within industry, they can be the foundation for a better management of the product development process. Due to this, we state that model based systems engineering forms the basis for systems engineering management to ensure, that project targets are achieved. In this contribution we point out, how a discipline-spanning system model can be used as the core of system engineering management: We introduce a modelling technique and its utilization for project planning as well as assessment and control. We present approaches for planning operational structures as well as technical reviews and further for measuring development progress and product maturity on different system hierarchy levels, using the information modeled within the system model.

Keywords:Model Based Systems Engineering; CONSENS; Systems Engineering Management;

1. Challenges within the Development of Complex Systems

The products of mechanical engineering and related industrial sectors, such as the automobile industry, are often based on the close interaction of mechanics, electrics/electronics and software engineering, which is aptly expressed by the term mechatronics. The conceivable development of communication and information technology opens up more and more fascinating perspectives, which move far beyond current standards of mechatronics: mechatronic systems having an inherent partial intelligence. These intelligent systems have the ability to react autonomously and flexibly on changing operation conditions. Further they are able to communicate and cooperate. The result is a network of intelligent systems.

* Corresponding author. Tel.: +49 5251 60-6267; fax: +49 5251 60-6268
E-mail address: Juergen.Gausemeir@hni.upb.de.

doi:10.1016/j.procs.2013.01.032

This opens up fascinating perspectives for the development of future mechanical engineering products. Besides these perspectives the development of intelligent systems also implicates some challenges. The noticeable increasing functionality of intelligent products, the cross-linking of elements within systems and their hardly manageable interactions induce a high complexity in the development process and make it challenging. Many engineering projects run into deep trouble, because they cannot manage these product and process complexity. As a consequence cost, time or quality targets fail. To challenge these increased complexity an effective cooperation and communication of the developers from different disciplines during the whole development process is required. An approach to manage these challenges is Systems Engineering and especially Systems Engineering Management.

2. Systems Engineering as a powerful Approach to Manage Product Development

Systems Engineering (SE) is a holistic and interdisciplinary approach to enable the realization of successful systems. It integrates systems thinking, discipline specific engineering approaches (methods, tools, and procedure models), human sciences, and management aspects [1], [2]. Systems Engineering's mission is to ensure 1) effectivity ("do the right things") and 2) efficiency ("do the things right") of product development. That means, that the task of systems engineering is to make sure that the right product is developed, with the aim to reach high quality at the first go with low costs. This is why there is not only a technical but a management component of systems engineering. Accordingly Systems Engineering is about looking at the "big picture" of product development [3].

Figure 1 illustrates the components of the system engineering approach according to HABERFELLNER et al. [4]. The approach describes product development as problem solving process. That can be distinguished into two components: system design and project management. **System design** addresses the basis design work. Core task is the creative development of the solution. This is where techniques like Model Based Systems Engineering (MBSE) are used. **Project management** dedicates to the organization and coordination of the problem solving process.

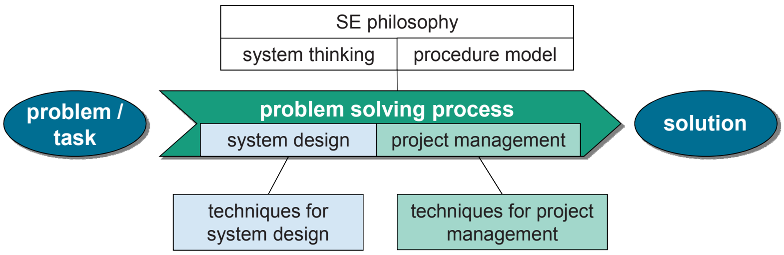

Fig. 1. Systems Engineering Approach according to HABERFELLNER et al. [4]

Product development can only be successful, if technical and organizational issues are considered. **System Engineering Management** brings both, system design and project management, together. It focuses on the engineering aspects of a project and addresses the following aspects: planning, assessment and control, risk management, measurement, decision management, configuration management, information management, and quality management. At this systems design provides the technical aspects respectively inputs; whereas project management provides the cost and time inputs [2], [3], [5].

According to the **5-layer model** the techniques and concepts of Systems Engineering (SE) and Systems Engineering Management can be used on different levels: product SE, project SE, enterprise SE, industrial SE, and socio-economic SE [6]. At his, each layer is an integrative part of the superordinated layer. As a consequence the product layer forms the basis for a holistic SE. As a consequence a lot of information for the other layers, have specified already at the product layer respectively in an overall product/system model. One example is, that project SE requires information about the technical and organizational interfaces for planning the development process.

Hence **Model Based Systems Engineering** (MBSE) becomes the basis for a successful system engineering management. It provides techniques for the holistic specification of a system model and can deliver all information, which are necessary to manage product development on the different layers. MBSE addresses the formalization of almost all models beginning in the conceptual design phase and continuing throughout development and later life cycle phase [7]. SysML (Systems Modeling Language) is the most famous example of the MBSE-initiative. This modeling technique is jointly developed from Object Management Group (OMG) and INCOSE. SysML supports specification, analyze, design, verification and validation of complex systems [8].

3. Specification of the Discipline-spanning System Model

For the specification of the discipline-spanning system model, we developed the model based specification technique CONSENS. Further we introduce the Mechatronic Modeller, which is a software tool supporting the specification of the system.

3.1. CONSENS: Conceptual Design Specification Technique for the Engineering of Complex Systems

CONSENS is preceded to the discipline-specific methods and complements them. It was developed within the CRC 614 to reduce the complexity of the development of complex systems [9]. From the engineer's point of view, this modeling technique provides a semi-formal notation of all necessary aspects for the specification of an complex system (Figure 2). The mentioned aspects are mapped on computer by partial models. Because the partial models are in relation with each other the principle solution consists of a coherent system of partial models describing the concepts of the product.

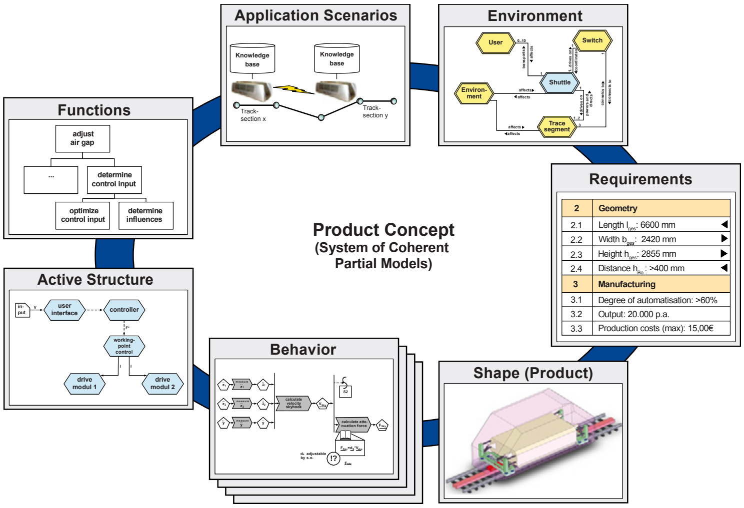

Fig. 2. Specification Technique CONSENS

Environment: This model describes the environment of the system that has to be developed and its embedding into the environment. Relevant spheres of influence (e.g. weather or superior systems) and influences (e.g. thermal radiation or wind energy) will be identified. The interplay between the influences will be examined.

Application scenarios: Application scenarios form first concretizations of the system. They concretize the behavior of the system in a special state and a special situation. Application scenarios characterize a problem, which needs to be solved in special cases, and also roughly describe the possible solution.

Requirements: This aspect considers the computer-internal representation of the requirements. The list of requirements sets up its basis. It presents an organized collection of requirements that need to be fulfilled during the product development (e.g. performance data). Every requirement is textually described and, if possible, concretized by attributes and their characteristics.

Functions: This aspect concerns the hierarchical subdivision of the functionality. A function is the general and required coherence between input and output parameters, aiming at fulfilling a task. Functions are realized by solution patterns and their concretizations.

Active structure: The active structure describes the system elements, their attributes as well as the relation of the system elements. It is the target to specify the structure of the system.

Shape: This aspect needs to be modeled because first definitions of the shape of the system have to be carried out already in the phase of the conceptual design. This especially concerns working surfaces, working places, surfaces and frames. The computer-aided modeling takes place by using 3D CAD systems.

Behavior: Two partial models are used to specify the behavior of the system. The partial model behavior – states defines the states of the system and the state transitions. The state transitions describe the reactive behavior of the system towards incoming events. The partial model behavior – activities describes the logical sequence of activities in the system. Especially, parallel executed activities and their synchronization can be described this way.

It is necessary to work alternately on the mentioned aspects and the according partial models although there is a certain order. The description of the environment, the application scenarios and requirements serve as the starting point. From the requirements the functions of the system are derived and arranged hierarchical. Based on the functions the active structure is modeled function oriented. That means, that the system elements are aggregated to modules according to their functional relations. The other way around there is a decomposition of system elements, which fulfills the corresponding partial functions. In the next step we model the shape of the system.

As a result we get the principle solution of the product. The partial models form the discipline-spanning system model. This system model is the starting point for the discipline-specific development of the product.

3.2. Mechatronic Modeller

During the development of the specification technique CONSENS it became clear, that software support is a necessity in order to be able to secure the overall consistency of the product and to manage its complexity. Hence, we developed the Mechatronic Modeller. The Mechatronic Modeller is a dedicated software solution, which is aligned with the specification technique. It offers a separate editor for each partial model. Figure 3 illustrates the user interface of this software tool.

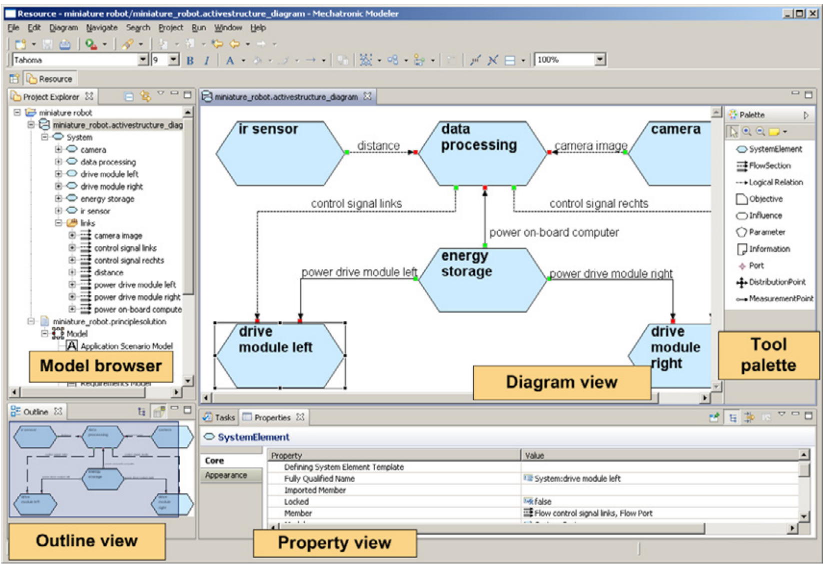

Fig. 3. Screenshot of the Mechatronic Modeller showing the Active Structure Editor

A so-called metamodel has been defined for the specification technique. It defines: which model elements are available during the description of the principle solution as well as how they are related to each other (abstract syntax) and how a model element has to be linked in order to have a meaning (static semantics).

The Mechatronic Modeller is based upon this metamodel. The product concept is computer-internally represented as a data model, which is the instance of this metamodel. In addition to the metamodel, following aspects of the specification technique had to be formally defined during the development of the tool: the meaning of a model element or a relation in context of a particular product concept (dynamic semantics), and how the models will be graphically represented (concrete syntax).

Since the Mechatronic Modeller is based on a metamodel, the overall consistency of the product concept can easily be ensured. In order to check it, it suffices to examine if the data model representing the product is compliant with the metamodel of the specification technique.

Due to being a dedicated software solution, Mechatronic Modeller addresses all particularities of the specification technique CONSENS. The very important aspect of usability can therefore be appropriately addressed. The aim is to hide the complexity of the model from the developer. Thus, several functions have been incorporated into the Mechatronic Modeller which support working with the specification technique and make the tool more comfortable and intuitive in usage. In particular, complex manipulations such as partial model reorganization by incorporating or deleting of hierarchy levels, are provided by the tool. Furthermore, cross-references between elements of different partial models are stored in the data model. Thus, Mechatronic Modeller is capable of handling complex dependencies between elements of different partial models within the system model. For example, it is possible to check which requirements have not yet been realized by functions or system elements. Requirements tracing is also possible, e.g. if a particular system element is to be exchanged, then the developer can examine which requirements had to be originally met by it.

In addition, the use of a metamodel opens up new possibilities for the analysis of the system model. For instance, the principle solution model could be formal verified using model checking and even simulated, provided all information necessary has already been included. Furthermore it is possible to analyze the system model according to get information for systems engineering management.

4. Management of the Product Development Process based on the System Model

In this section we explain how to get and use the information modeled within the system model to manage product development. We begin with a short description of the development process of complex systems and introduce product development as a complex system. Then we will present how the system model can be used for systems engineering management. We present how to plan the structure of the development process, how to align the technical and organizational view on the development process, and how to plan and monitor the development progress using the information modeled in the discipline-spanning system model.

4.1. Development Process of Complex Systems

On the highest degree of abstraction, the development process of intelligent systems can be subdivided into the discipline-spanning conceptual design and the discipline-specific development (Figure 4). Within the conceptual design, the basic structure and the operation mode of the system are defined.

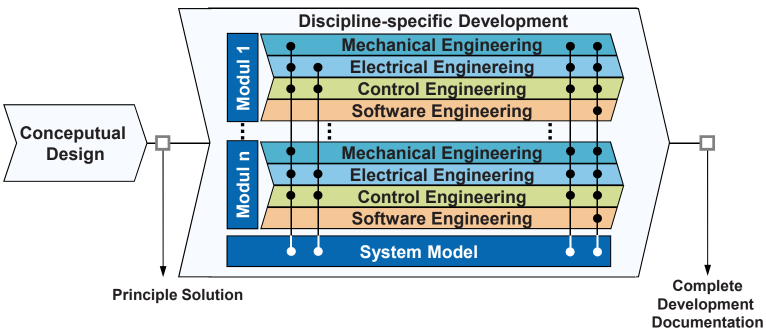

Fig. 4. Development Process of complex Systems

All results of the conceptual design are specified in the so-called principle solution. By using CONSENS, the system that is to be developed will be described in a holistic, discipline-spanning way. The description of the principle solution provides all relevant information for the structuring of the system and forms the basis for the communication and cooperation of the developers from different disciplines. Based upon the principle solution the subsequent discipline-specific development is planned and realized. The aim of the discipline-specific development is the complete description of the system by using the construction structure and the component structure. In so doing, all defined modules are developed in parallel, and each module is developed in parallel in the participating disciplines.

4.2. System of Product Development

The information modeled in the overall system model can be used for the management of the product development process. This is possible because there is a direct dependency between the structure of the system model and the structure of the development process. Figure 5 depicts this central position of the system model for the management of product development.

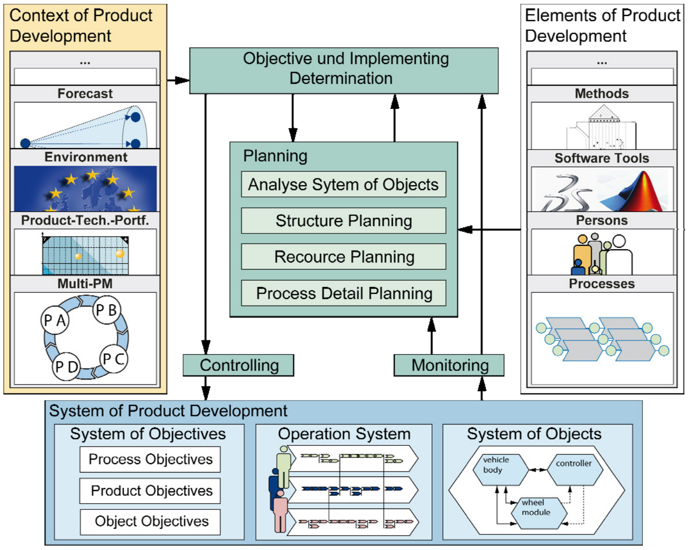

Fig. 5. Systematic Description of the System of Product Development and its embedding in the Control Loop of Project Management

Based on systems theory and according to ROPOHL [10] and NEGELE [11] we regard product development as a complex system. The **system of product development** consists of three interrelated systems: system of objectives, operation system, and system of objects (blue area in Fig. 5).

The **system of objectives** exists of product, process and object objectives. These objectives are derived from the environment of the product development and are adapted according to the progress of product development. Time, cost or quality objectives can be changed according to the process or technical performance.

Within the **operation system** the elements of product development are configured and tailored according to the system of objectives and to the system of objects. The elements of the operation system (e.g. engineers, methods, software tools, production machines) transfers the product objectives into objects (e.g. CAD-models, test results, prototypes, physical products) taking the time and cost objectives into account.

The **system of objects** contains all outputs of a development project. That means virtual objects (e.g. CAD-models, virtual prototypes) as well as physical objects (e.g. physical prototypes, parts). Also the system model is part of the system of objects.

For the **management of product development** the system is embedded into the classic control loop of project management (green area in Fig 5). This covers planning, monitoring, and controlling.

The starting point for **planning** the development process is the system model. This is analyzed in order to get all information (e.g. modules of the system, involved disciplines), which are necessary to plan the structure of the development process. Further the necessary development processes are identified. The next step is the allocation of resources (e.g. engineers, methods, software tools, test stands, and machines) to the processes of the operation system. The last step for planning the development process is the scaling of the processes as well as the supporting methods and tools, according to the ability of the engineers.

For **monitoring** the development progress the process performance and the technical performance are measured. To measure the efficiency the operation system is monitored and compared to the time and cost objectives; for the effectivity the object system and its maturity is compared to the quality objectives.

Within the **controlling** process actions to reach the time, quality, and cost objectives are identified and implemented. This spans either the adjustment of the operation system or the adaption of the objectives.

4.3. Planning the Structure of the Development Process

The complexity of intelligent systems and their development processes can be handled by defining modules that form logical and functional units, which can be separately developed, tested and maintained. The procedure to define these mostly independent modules is integrated in the conceptual design phase [12].

During the conceptual design phase the partial models of the principle solution are analyzed against the backdrop of product structuring. For the structuring different methods are applied that need different information. The relevant information for the application of these methods are extracted from the partial models.

For analyzing the information the Design Structure Matrix (DSM) by EPPINGER et al. [13] is used. It enables the analysis of the interrelations of the system elements. The relevant information of the system elements' connections are mainly extracted from the partial models active structure and shape. For the structuring two extreme views on the system are created. One focused on a shape-oriented structure and one focused on a function-oriented structure. For the further refinement of the product structure the Module Indication Matrix (MIM) by ERIXON [14] and its extension by BLACKENFELD [15] are used. The MIM allows to take into account the properties of the system elements and to summarize them according to matching aspects. Input information are for example the realized functions, the used material, or maintenance intervals of a system element. The result is the development-oriented product structure. It integrates the two basic and mostly contradictious views of a shape- and function-oriented product structure.

The resulting development-oriented product structure is described by two hierarchical trees. On one axis the shape-oriented structure is described. The other axis shows the function-oriented structure. The assignment of a function to a system element and vice versa is described by relations. Figure 6 illustrates the structured active structure of an innovative railway system so called "Neue Bahntechnik Paderborn/RailCab" (http://www.nbp.uni-paderborn.de).

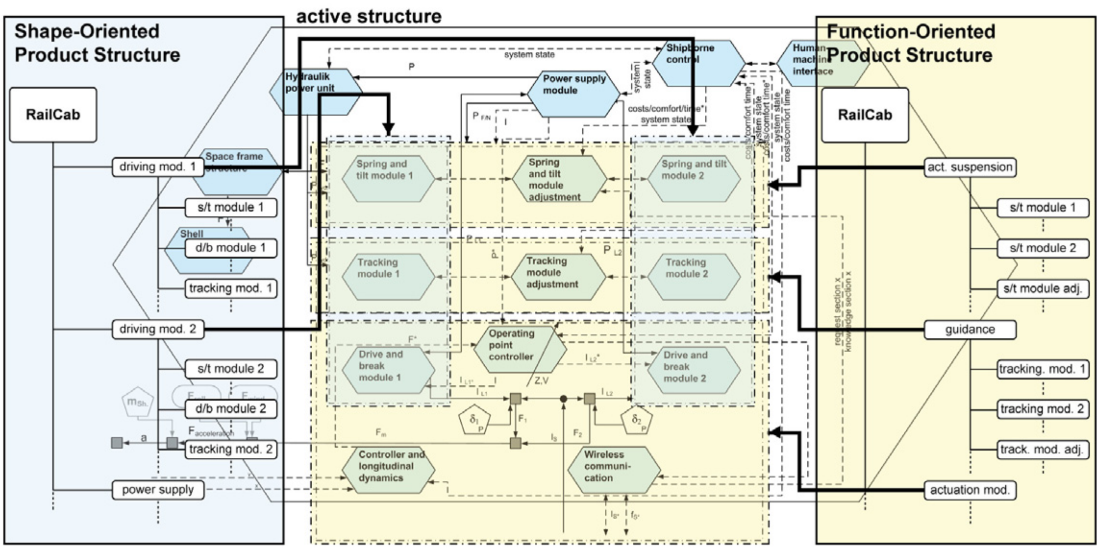

Fig. 6. Initial Shape- and Function-oriented Product Structure

Based on the product structure the development process is planned and for each module from the shape oriented product structure a swim lane is scheduled. In order to address the involved disciplines this module-specific development swim lane is detailed in discipline-specific swim lanes. Therefore we specified additional information within the active structure like disciplines, responsible for a system element or module. The swim lanes contain the main development tasks for the generation of the objects and first discipline- and module-spanning synchronization points. These technical interfaces are derived from the functional interfaces within the product structure.

4.4. Alignment of Technical and Organizational Synchronization Points

Beside the technical interfaces, it is necessary to plan the organizational synchronization points. Therefore we adopted the approach form SOSA [16] for our specification technique CONSENS. It is possible to derive automatically the necessary information for the alignment of technical and organization synchronization points form a system model, specified with CONSENS and the Mechatronic Modeller.

The basic idea is, that the development teams have to communicate more effective during the development process. In order to achieve this, the organizational communication and synchronization points should be planned on basis of the technical interrelations of the technical system. Using the approach from SOSA it is possible to map the technical and organizational interfaces (Figure 7).

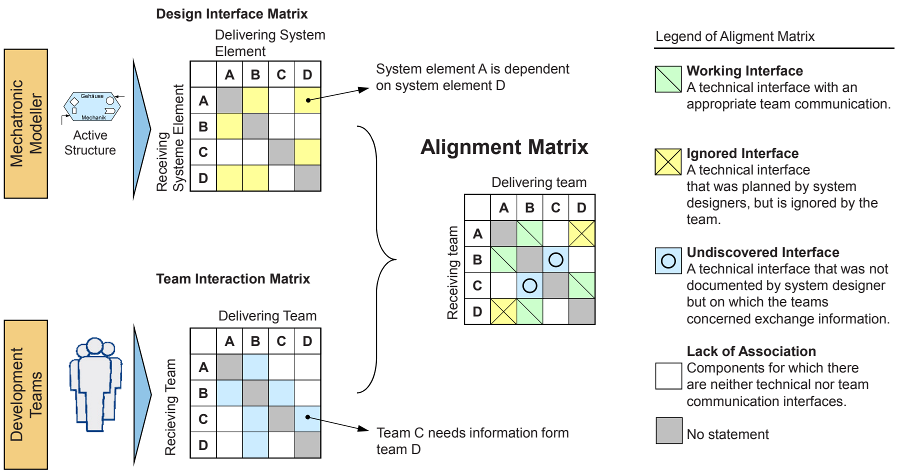

Fig. 7. Alignment of Technical and Organizational Synchronization Points according to SOSA [15]

First, the technical interfaces of the system are identified and documented in the design interface matrix. Therefore we use the information, modeled within the active structure of the system model. Second, the team interactions are described within the team interaction matrix. Therefore the technical inputs and outputs (e.g. documents, CAX-models, information about technical interfaces), which are expected from the involved engineers, are documented. As a result of the comparison of both matrixes we get the alignment matrix. Thereby missing and unknown, or even ignored interfaces could be identified. As a consequence the systems engineer or the project manager can derive activities for the improvement of the communication of design teams or individual engineers. Vice versa responsibilities can be allocated to system elements or modules using the team interaction matrix.

4.5. Product Maturity based Planning of the Development Progress

We get the technical and organizational communication and coordination points so far. In addition a time sequence for these synchronization points is required. Therefore we define the technical reviews.

The basic idea is to ensure, that development progress rises continuously over development time. Hence the aim is to establish a balanced plan with respect to schedule, technical performance and quality based on the technical description of the product in the system model.

For planning and controlling the development progress we use the product maturity level as one key performance indicator (KPI). We utilize product maturity as a KPI to measure the module- and domain-spanning development progress, because it provides the possibility to measure process performance and technical performance in an integrative way.

The relevant information for planning the development progress are extracted from the system model and project management: 1) From the system model requirements, system elements, modules and the involved disciplines are extracted. Further it is necessary to know the objects (development results), which have to be developed. 2) From project management the milestone checklist and the validation plan are required. The milestone checklist delivers all necessary milestones and quality gates as well as the assigned target maturity levels of parts, components or modules etc. The validation plan provides all required valuation procedures according to the characteristics of the product (e.g. expert estimation, computation, simulation, testing on a virtual or physical prototype). The valuation procedures are used to ensure product quality. Therefore product properties are compared to requirements by using suitable valuation procedures. It is important to know, that each valuation procedure has an uncertainty factor, which have to be considered at the maturity calculation. For example the uncertainty of an expert estimation could be 70 %, while the uncertainty of a test of a physical prototype could be 5 %.

With the information getting form the system model and project management a Multiple Domain Mapping Matrix (MDM) is filled (Figure 8, left). The MDM itself consist of several Design Structure Matrices (DSM) and Domain Mapping Matrices (DMM).

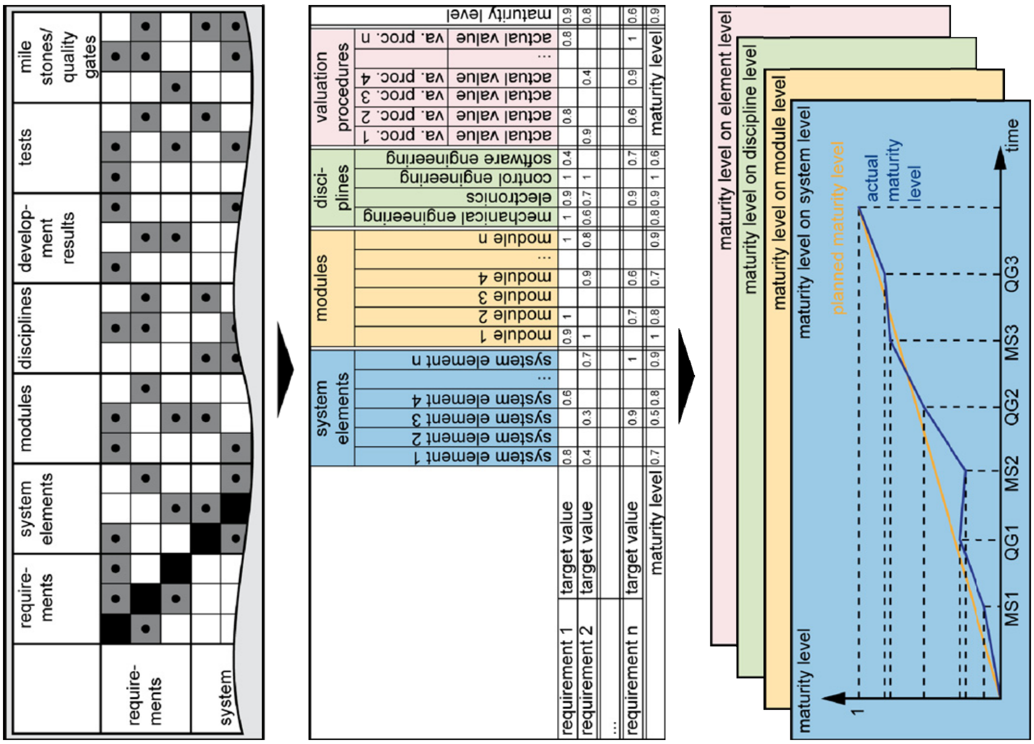

Fig. 8. Planning and Monitoring Development Progress using Product Maturity as a KPI

For planning the development progress two steps are necessary: First, the DSM of the development results have to be restructured. The output is a logical sequence of the objects, which considers all interrelations of the objects from the involved disciplines. Second, we structure all the other DSMs and DMMs considering the restructured DSM of objects. As a result we get a logical sequence of milestones/quality gates with corresponding objects and their target maturity level.

The use of the information modeled in the system model and their transformation within the MDM opens up the possibility to calculate the product maturity on system level, module level, domain level, and system element level. (Figure 8, middle). The necessary information for calculating the product maturity level we get e.g. from product data management, or virtual and physical tests at the planned design reviews. Through a comparison of the target maturity level and the actual maturity level the development efficiency can be evaluated (Figure 8, right).

5. Summary and Future Work

In our contribution we present the specification technique CONSENS and its utilization for systems engineering management. We use the information of the system model for planning and monitoring the development process of complex systems. We have introduced a basic concepts for a) planning the development structure, b) the alignment of technical and organizational communication and coordination points, and c) the planning and controlling of the development progress using the information modeled in the overall discipline-spanning system model.

Therewith we pointed the importance of MBSE for Systems Engineering Management; MBSE provides the necessary information and forms the basis for Systems Engineering Management.

Our future work will focus on: Planning a) detailed operational structure, b) organizational structure, c) detailed technical interfaces and d) development teams using the information of the overall system model. Further we will focus on probabilistic planning of the system of product development. Therefore we will extend CONSENS as well.

Acknowledgements

This contribution was developed in the course of the Collaborative Research Centre 614 “Self-Optimizing Concepts and Structures in Mechanical Engineering” (Speaker: Prof. Gausemeier) funded by the German Research Foundation (DFG) under grant number SFB 614.

References

1. International Council on Systems Engineering (INCOSE): Systems Engineering Handbook – A Guide for System life cycle processes and activieties. International Concil on Systms Engineering (INCOSE), Version 3.2, 2010.
2. Body of Knowledge and Curriculum to advance Systems Engineering (BKCASE): Guide to the Systems Engineering Body of Knowledge (SEBoK). Version 1, 2012.
3. NASA: NASA – Systems Engineering Handbook, NASA/SP-2007-6105 Rev1, 2007.
4. R. Haberfellner, E. Fricke, O. Weck, S. de Vösser: Systems Engineering, Orell Füssli Verlag, Zürich, 12. Auflage 2012.
5. A. Kossiakoff, W. N. William: Systems Engineering – Prinzicples and Practice, John Wiley & Sons, West Sussex, 2003.
6. D. K. Hitchins: Systems Engineering: A 21st Century Systems Methodology. John Wiley & Sons, West Sussex, 2007.
7. INCOSE Systems Engineering Vision 2020, 2007.
8. S. Friedentahl, A. Moore, R. Steiner: A Practical Guide to SysML: The Systems Modeling Language. Elsevier, Oxford, 2nd Edition, 2011
9. J. Gausemeier, U. Frank, J. Donoth, S. Kahl: Specification Technique for the Description of Self-Optimizing Mechatronic Systems. Research in Engineering Design, Vol. 20, No. 4, 2009, Springer, London.
10. G. Ropohl: Einleitung in die Systemtechnik, Systemtechnik – Grundlagen und Anwndung, Hanser, München, 1975.
11. N. Negele: Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung. Dissertation, Fachgebiet Raumfahrttechnik, Universität München, Herbert Utz Verlag, München, 2006.
12. D. Steffen: Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation, Faculty of Mechanical Engineering, University of Paderborn, 2006, HNI-Verlagsschriftenreihe, Vol. 207, Paderborn.
13. S. Eppinger, D. Whitney, R. Smith, D. Gebala: A Model-Based Method for Organizing Tasks in Product Development. Reserarch in Engineering Design, Vol. 6, No. 1, Marc, 1994, Springer Verlag, London.
14. G. Erixon: Modular Function Deployment – A Method for Product Modularization. Dissertation, Royal Institute of Technology, 1998 KTH, Stockholm.
15. M. Blackenfeld: On the Development of Modular Mechatronic Products. Licentiate Thesis, Royal Institute of Technology, 1999, KTH, Stockholm.
16. M. E. Sosa, S.D. Eppinger, C.M. Rowles: Wie Sie ihre Ingenieure zum Reden bringen. In: Harvard Business Manager, Dezember 2007