

Chapter 2

Product and Systems Engineering/CA* Tool Chains

Kristin Paetzold

Abstract For the development of interdisciplinary technical systems such as CPS, systemic approaches which stringently summarize the logic of development are currently available. These approaches are suitable to support the complexity of both the CPS as well as the related developmental processes. However, these development methods are relatively generic. An adaptation or a tailoring to specific conditions of both the products under consideration as well as the development of boundary conditions is absolutely necessary to use them effectively and efficiently. For the development of CPS also a variety of IT tools which effectively support the product development but only if they are well coordinated with the corresponding processes, are already available. If the interfaces are described sufficiently and comprehensively, and the data characteristics of the results of the various development activities are taken into account, media discontinuities can be reduced. The major challenge in the development of complex technical systems is the overall system analysis and the system integration. To this end, modern methods such as model-based engineering in general and model-based Systems Engineering in specific, provide powerful approaches that must be applied and adjusted for the purposes of the product and process characteristics. This adjustment process to product development and the integration of MBSE approaches into the IT-structures may be seen as the main challenges for the future.

Keywords Product development • IT-structure • Systems engineering • Data- and information flow • Model-based systems engineering

2.1 Introduction

Developing Cyber Physical Systems (CPS) whose functionality is caused by strong interactions between physical and computational components (Sztipanovits 2007), pose major challenges to the development and especially the design of development

K. Paetzold (✉)

University of German Federal Armed Forces (UniBW), Munich, Germany

e-mail: kristin.paetzold@unibw.de

processes. CPS include solution approaches from various engineering fields such as mechanics, electrical engineering, computer science, control engineering but also thermodynamics or materials engineering. The system behaviour can ultimately no longer be derived just as a sum of individual partial functions. Instead, synergies can be skimmed off by the diverse interactions of sub-functions. This product characteristic results in a number of challenges for both the process itself as well as the methods and tools used in the development process.

Processes in general as well as product development processes in particular can be understood as a series of interrelated activities that give rise to a valuable result for the company (Hammer 2001). For product development, these activities can be specified as that they include all operations, from the product idea to the start of production (as Ehrlenspiel and Meerkamm 2013). Development processes are also characterized by a certain uniqueness. It is not necessarily the aim of achieving an always equal result but rather finding a customized solution to specific customer requirements and operating conditions, which is characterized by a high level of functionality at an equivalent quality. This leads to a paradox: in a company is never expected the same exact result of a development process is never expected twice, which is associated with the fact that the development processes are different in each case. Nevertheless, not only for reasons of efficiency and effectiveness it requires clear procedures in the design of the development processes which are connected to a standardization of these. In addition, development processes are distinguished by a high degree of innovation and creativity (Kline 1995), which again the mentioned paradox supports.

Basically, product development processes can be characterized by the following four characteristics (see Fig. 2.1):

- Data and information about products in general and to CPS in particular arise only in the context of development. In order to still be able to work

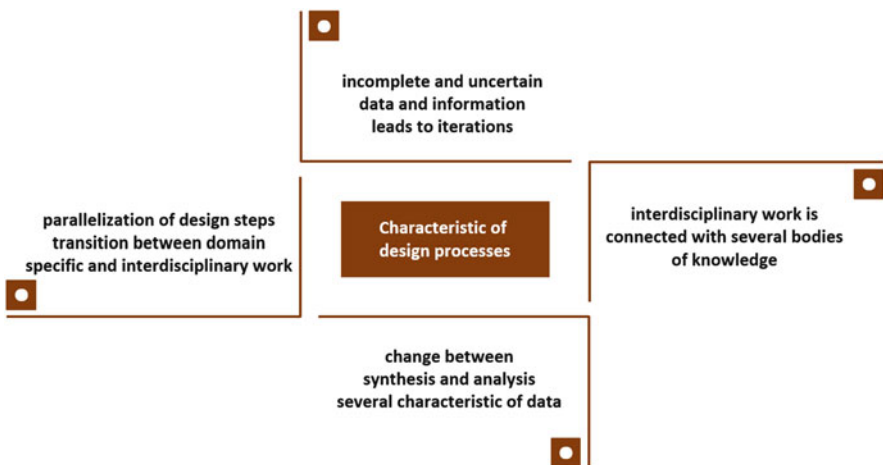


Fig. 2.1 Characteristics of the product development processes

result-oriented, it is common practice to make assumptions first which need to be concretized and evaluated in the later stages. Development processes are in accordance with the fact that there must be dealt with uncertain and incomplete data and information (Freisleben and Schabacker 2002) that become reality just in the course of development.

- The development of CPSs is highly interdisciplinary. Therefore, the different domains need to cooperate closely in all stages of the development. The challenge is that in the different domains, different models are used for the development of (Horvath and Gerritsen 2012), which in turn differ from the model approaches of system integration. The models describe the same technical system with different perspectives on it, what leads to a high variance in the information content of models.
- In terms of a concurrent engineering, developments in the individual departments run parallel. For the domain-specific development, data and information from other departments are required in general. This requires individual activities and tailored interface management to ensure that the data and information are available with sufficient quality at any stage of development.
- Development processes consist of a permanent exchange between analysis and synthesis. Data and information which are defined as part of the synthesis, need to be investigated and assessed by appropriate analytical steps regarding the fulfilment of requirements, which in turn may lead to corrections if there is the need. This is connected to the thought, that the characteristics of data for synthesis and analysis can be distinguished (Weber 2005).

The development processes of complex technical systems as CPSs thus require complex processes to secure the expectations in terms of functionality and quality. Such properties result in the fact that iterations are essential during the development. Concurrency of activities and the strong links between these individual activities through data and information exchange lead to pronounced nonlinearities in the development processes. These nonlinearities are only accessible through a detailed view of the data and information flows within the development but need to be taken into account in the design of processes.

In defining of development processes in companies, at least the corporate knowledge is manifested in the designed technical systems. The approach in the development is the result of an evolutionary process which does not only take the product of evolution into account but also effects the “Lessons Learned” or historical data and information. Reversed, development processes will be anchored in the definition of departments, team structures and responsibilities, which are in turn the basis for associate -processes such as release decisions or the actual project execution. Furthermore, emerge from the descriptions of the development process for CPS data and information requirements for each activity that must be ultimately provided and managed by the available IT infrastructure.

Process models for development are therefore not only used to represent the inherent logic in the development but also serve as basis for the design of the organizational and operational structure in the development departments and the

design of the IT environment in enterprises. They also form the basis for associated processes such as risk management, change management, or the verification and validation management, which are often implemented as part of the division of labour as independent tasks of product development. Therefore, models for the description of the development process have to be considered at three levels also. Generic process models which reflect the logic of development, proved to be just little helpful for the practices of process design and optimization, because they are coarsely granular. It requires a context-specific refinement and adaptation of process models to the specific conditions in the company, to the market or the industry and, ultimately, to support the development in terms of workflow processes. In order to achieve this aim, the data and information flows are analysed within the development to link targeted activities with each other. This relatively fine-grained level of process description appears necessary to assess uncertainties due to incomplete and uncertain data, in order to increase the product maturity with the initiated activities and thus to avoid unnecessary iterations. Processes on such a fine-granular level are mainly characterized by detailed interface descriptions which have to take three aspects into account:

- Procedural interfaces result from the logical sequence of development steps which contribute to the increase of product maturity;
- Organizational interfaces define responsibilities for process steps or release mechanisms, and thereby support the quality assurance during the development;
- Formal interfaces link the IT tools which are used during the development in order to secure consistent data and information flows.

A closer look at these three levels of process description, the underlying models and the methods and approaches to each process support will be taken in the following subsection.

2.2 Generic Procedures for the Development of Interdisciplinary Products

Cyber Physical Systems are not only distinguished by the wide range of functions but also by the strong interlinkage between those functions. This distinction guarantees the variability in the response of the system in different states respectively its flexibility and robustness of the systems behavior. Such complex systems require adequate procedures in development. Systemic approaches will be used to make the strong dependences between systems design and process design explicit.

Haberfellner et al. (Haberfellner et al. 2015) recommend to organize the processes based on models for the system description, which map again both functional and structural aspects of the system being to designed (Fig. 2.2). Such holistic approach forms the basis of today's popular procedural models. System development itself is based on defined requirements of the overall system, which are

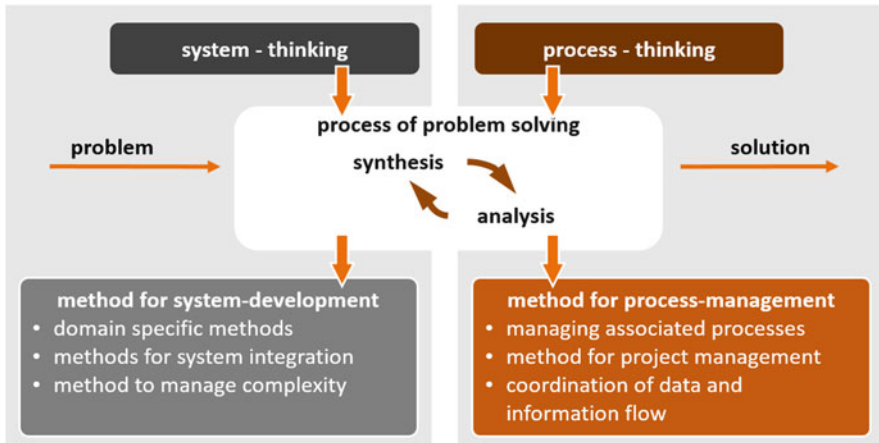


Fig. 2.2 Systemic thinking in the development of complex technical systems (according to Haberfellner et al. 2015)

successively decomposed to smaller units, where solution approaches are needed (from rough to detail). This approach in a phase structure (macro logic) with the basic steps planning, designing, and finishing of the results (e.g. Pahl and Beitz 2007). This phase structure in turn needs to be put in concrete terms, which depend on the system structure and the area of expertise, in which the development takes place.

Within these process models, the main task of the development is to seek for solutions and to check them with regard to the requested performance. This problem-solving process (micro logic) (Ehrlenspiel and Meerkamm 2013) manifests itself in a permanent alternation of synthesis and analysis and can be deduced from typical approaches to problem solving of individuals.

Individuals as developers need methodological support in the development from two perspectives: on the one hand, methods for system design and to control the system's complexity are needed and on the other hand, methods for process support and coordination of the development tasks are required. Both perspectives are explained briefly below.

2.2.1 Micro-logic in Development

The micro-logic in the development describes the operations at the level of concrete project work (Gausemeier et al. 2004) and supports the systematic processing for solving partial of problems within individual phases of the development process. The basis for this processes are generic procedures of psychology of thinking (Miller et al. 1973).

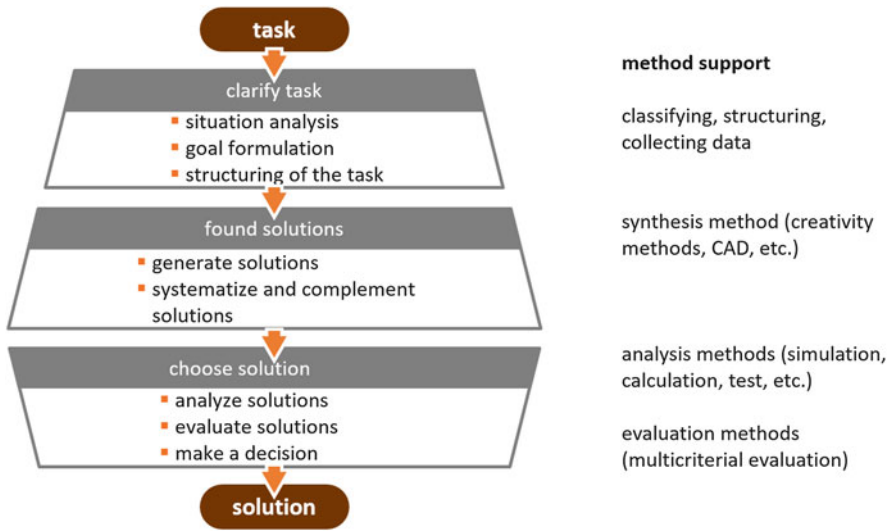


Fig. 2.3 Problem solving cycle according to Ehrlenspiel (Ehrlenspiel and Meerkamm 2013)

The problem-solving cycle (according to Ehrlenspiel and Meerkamm 2013) is shown in Fig. 2.3. After defining the roles, the broadest possible solution space is created within the framework of the synthesis which is then analysed and evaluated with respect to the achievement of objectives. Thus, the solution space is always limited. This constant interplay of analysis and synthesis is ultimately one of the reasons for iterations in the development process.

The product life cycle determines some implications for the coordination of data and information flows as well as the tool integration in the development. The individual steps can be associated with categories of methods as it is illustrated in Fig. 2.3. Special attention is given on the methods and tools for synthesis and analysis. The two process steps are interconnected via data and information flows (Fig. 2.4). It is crucial that for each of the two steps different categories of data must be captured.

For this purpose, two categories of data must be distinguished (Weber 2005):

- Characteristics which define the product; they are defined by the developer and thus serve as an adjusting screw to manipulate the properties.
- Features which describe the product behaviour; the properties cannot be influenced directly by the developer.

Exemplary for this mind set is the rough designing and dimensioning in the design. In order to meet the properties' conditions several components, for instance, part lengths or materials (characteristics) with respect to the strength or weight of the part have to be defined which must fulfil specified functions (features) (Weber 2005).

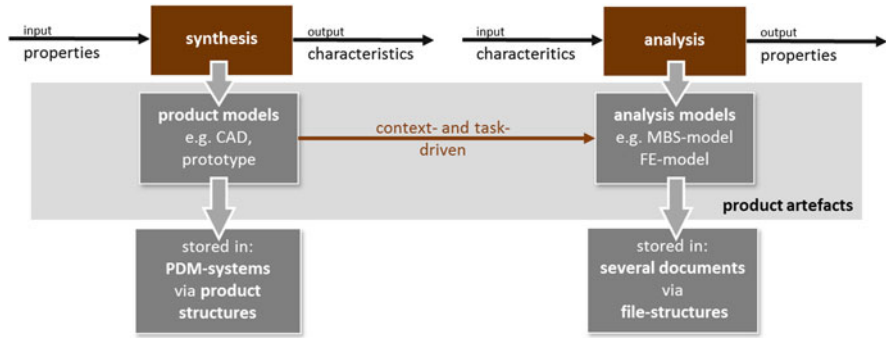


Fig. 2.4 Information flows in product development

Input values of the synthesis are properties that are requested by the technical system. Those values it is necessary to find corresponding characteristics. By analysing those values characteristics are examined to determine whether the property's performance is guaranteed or specified characteristics support the functional performance. Consequently, properties can be divided into target properties, from the customer requirements and actual properties which are the result of a development step.

Which of the various methods is suited best, is on the one hand determined by the objectives of the analysis and on the other hand by the product's maturity or rather by the developmental progress and its associated data quality respectively the uncertainty regarding the data (Reitmeier 2015). This implies the need to distinguish between product models as a result of the synthesis and analysis models. Result of synthetic steps are product models such as CAD models, prototypes or test setups. During the analysis it is necessary to transform the product models into analysis models in order to make them accessible for calculations, simulations and experiments. Examples of analysis models, such as FE models, Matlab Simulink models or test body illustrate the diversity which needs to be handled. Below both categories are summarised to product artefacts. Such a distinction is necessary to understand for instance breaks in the data and information flow.

The resulting problem solving cycle (Ehrlenspiel and Meerkamm 2013) as approach of process description on the problem-oriented level is therefore non-specific, which means that it is equally suitable for all disciplines. The problem-solving cycle is primarily used to provide situation-specific methods. Resulting instructions have descriptive character. It addresses the question of HOW the solution finding should be done (Lindemann 2007).

2.2.2 *Process Models as Macro-logic in Development*

Macro-logic in development is described by process models in product development. These process descriptions address the logical and chronological order from the idea to the finished product. These physical issues are relevant for the actual characteristics of the individual activities. This is the best reason why the process model differ quite domain specific. They follow a prescriptive nature and deal with the question of the WHAT within in the Process (Lindemann 2007).

Process models, as phase concepts, have a high degree of generality even within the domain. They require a sector-specific adaptation and concretisation for being used as process basis in terms of project management. It is their big advantage, that the development process is divided into manageable sections which partly can be found in the operating structures of the development departments in the company again. The structure is classified in two ways (Lindemann 2007):

- **Logical:** During development, an abstract situation will be concretized, whereby the data and information base progressively will be completed. Thus, the adaption for the methods which are used for synthesis and analysis is needed.
- **Time:** The order of the process steps and the details of the data base can therefore be used for project planning.

Process models describe generally how a predefined target system can be transferred to a more or less abstract level in a specific technical system (Negele 1998). Process models vary depending on the technical domain in which solutions for technical systems are being developed, because this solution finding and specification are very strongly influenced by the used physical effects and relationships. Nevertheless, independent of the considered domain, four phases can be identified at a very generic level (Fig. 2.5), which remind of (Pahl and Beitz 2007).

Planning Phase: This phase conduce the definition of the task and of the demands on the system to be developed. In addition to the customer or user requirements it is necessary to consider the affecting or limiting constraints from both perspectives, the company's internal situation as well as from the environment or the use out. The aim of this phase is to prepare all the development factors influencing, so that therefrom parameters for the development itself can be derived. Results besides the requirement specification is the definition of the system purpose and the expected system behaviour.

Concept phase: This phase concerns a system decomposition based on the system purpose and the expected system behaviour (Andreassen 1980). The related detailing is accompanied by a permanent exchange between function synthesis and synthesis principle. Function definition and refinement are always associated with the search for fundamental solutions, which can be derived from the use of physical effects. The choice of physical effects in turn influences the further breakdown into sub-functions (Ponn and Lindemann 2011). These partial solutions are assembled to form a working structure, which in turn defines the basic structure of technical systems.

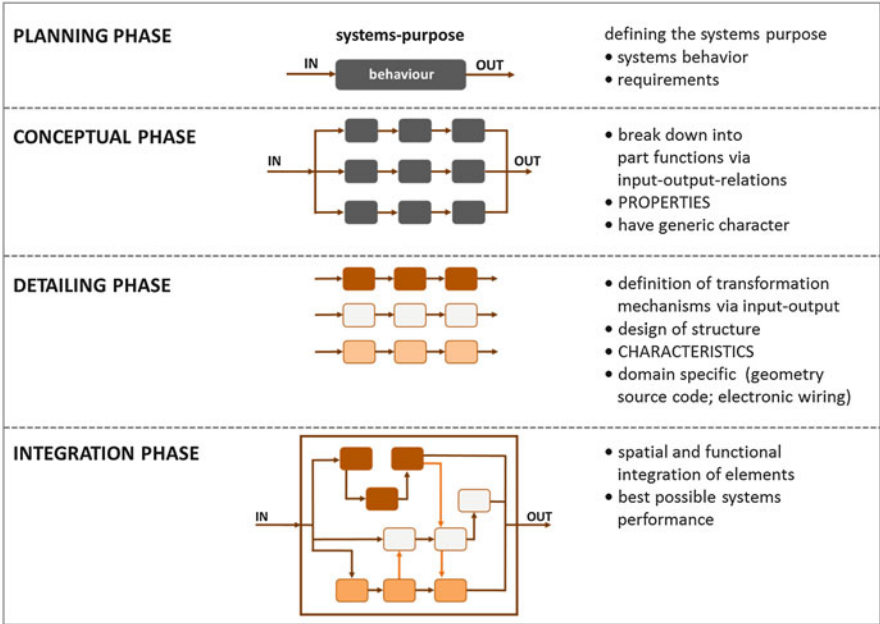


Fig. 2.5 Principle stages in product development

Design phase: The design phase is mainly influenced by the fact that the individual functions and fundamental solutions are specified and refined. Depending on the size and complexity of the technical system to be developed this design work is carried out in several different teams who then are specialized in the development of individual components.

Integration phase: Since today also in the domain-specific development, a high level on the division of labour can be observed, the integration of the results into an overall system is becoming increasingly important. In this context takes place both, a spatial and functional integration. Adaptions of the integration adjustments to the partial results or components are made such that the overall system then has the best performance. In and of itself it does not make sense to consider the system integration as downstream stage in development. Both in terms of efficiency as well as from a qualitative and functional point of view out a parallel consideration during development is strongly preferred. While this is quasi still immanent in the concept phase, it is challenging especially for further detailing.

Depending on the specific domains, different strengths of effort are required for each stage in development, which results from the physical relationships and the mathematical description associated. While the construction has to rely on the use of geometry and materials for continuous-time descriptions, in development of integrated circuits e.g. for the description of individual components, discrete event

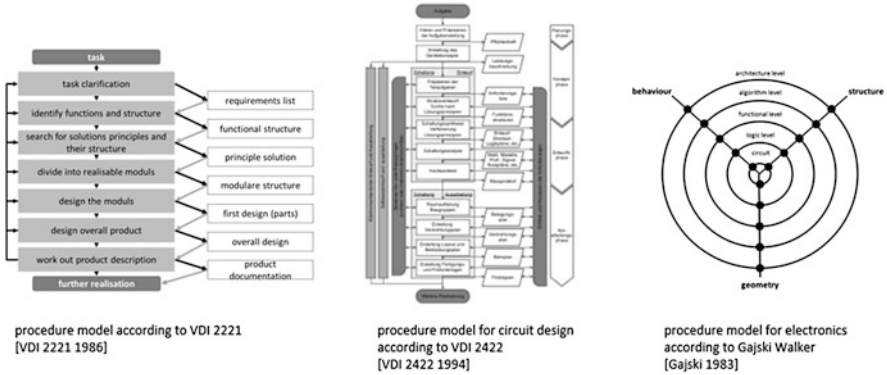


Fig. 2.6 Examples of domain-specific process models

approaches can be used. As a result, at a certain level the development is automated in circuit design, which is not feasible for mechanical design.

Differences in the specialized domains also arise because for function descriptions there is a cross-domain understanding, but the structure depths in the domains are very different (geometry in engineering, integrated circuits in electronics, source code in software development).

For these reasons, the process models in the specialized domains are correspondingly varied. Fig. 2.6 shows examples of procedures from the domain engineering, electrical engineering and software development. The differentiation of the domains is not only reflected in the structural form but also in individual activities and their designations. A detailed compilation of domain specific process models can be found in (Eigner et al. 2014).

Process models provide first only a sequential series of more or less detailed process steps, following the logic “from the abstract to the concrete”. In the development of this and especially as a result of adaptation to the specifics of different disciplines to the process models has been attempted to consider the general character of development processes. Correspondingly to the structure of the process models three main types can be differentiated. The simplest structure is a sequential approach as “logic-beam”. Partly, this structure is also called waterfall model. Since the need of iterations is well known, this was integrated into the waterfall model (Fig. 2.7a). Exemplary adaptations can be found in the VDI 2221 (VDI 2221 1986), the approach to circuit design according to VDI 2422 (VDI 2422 1994), generally waterfall model of software development (Boehm 1979) but also in the Y-chart to Gajsky-Walker, which takes place in the electronic development (Gajski 1983; Walker and Thomas 1985). In the latter, there are three logic-rays that reflect and merge the three typical ways of looking at an integrated circuit.

By the logic beam is folded over at the level of activities of the draft to a V-model (Fig. 2.7b), one can point out the importance of the property protection for development results. In terms of the function of protection and a high quality product

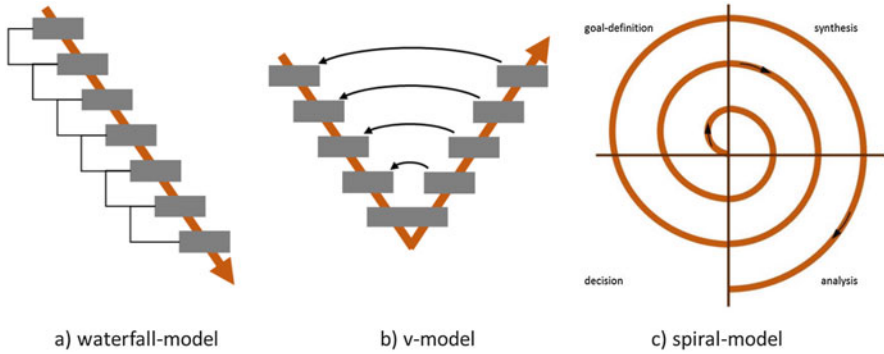


Fig. 2.7 Structures of process models

as by increasing demands in terms of system reliability, verification and validation of technical systems is becoming increasingly important. This type of visualization and implementation supports the way of thinking. V-model approaches can be found in software development (Forsber and Mooz 1991), but especially for the description of the development of complex technical systems as in mechatronics (VDI 2206 2004).

Another structural adjustment is the spiral model (Fig. 2.7c), which is more commonly used in software development (e.g. Boehm 1988). Ultimately, the logic-beam thereby will be “wound”. This form of modelling illustrates not only the paradigm of successive refinement (NASA 1995). The introduction of the quadrants also addresses and integrates the logic of the problem solving cycle with the phases tasks careful synthesizing, analysing, evaluating and deciding that must be passed through to each stage of the procedure. This way of thinking are as the application of agile methods in software development as a basis. In this way of thinking is based e.g. the application of agile methods in software development.

2.2.3 *Process Models for CPS as an Interdisciplinary Technical System*

To explain the procedure for developing CPS as interdisciplinary technical systems, domain-specific approaches are not appropriate. But the logic illustrated with the four phases, needs to be considered due to the inherent systemic view. In considering such interdisciplinary products there are other aspects of importance:

- The complexity of CPS is not based solely on the number of elements (functions and components) but also on the cross-linking diversity. The structural characteristics of individual sub-elements are very heterogeneous, with the result

that interfaces can be partly difficult to detect and to interpret. Emergent system behaviour of CPS can be the result.

- The expected from CPS behaviour is more diverse. This is not simply because the CPS must adequately respond to different environmental conditions, but also to its own system states. New business models, offering services in the context of the CPS and the requirement for an intensive consideration of the product life cycle mean that different stakeholders have different demands on the system behaviour or expect a specially specified behaviour from CPS.
- The division of labour for developing a CPS presents itself much more heterogeneous than in conventional technical systems. Many companies cannot hold the entire technological know-how for the solutions. In some industries, significant shifts in the leading competencies of the company can be observed. Core competencies are no longer alone in the mastery of specific technologies but in system integration. In addition to intensive cooperation with suppliers, components of the shelf (COTS) gain importance. This requires special attention to the problems associated with the product development processes, such as requirements management, risk management or the security management.

For the development of interdisciplinary technical systems like CPS the V-model has been proved as structural approach (Fig. 2.8). This reflects not only the four phases shown in the preceding chapter, but also explicitly address aspects such as property protection and modelling aspects. Thereby, not only the substantiation of the development results is discussed but there are also first indications for the description and analysis of data and information flows given.

For each phase now the areas of responsibility need to be expanded.

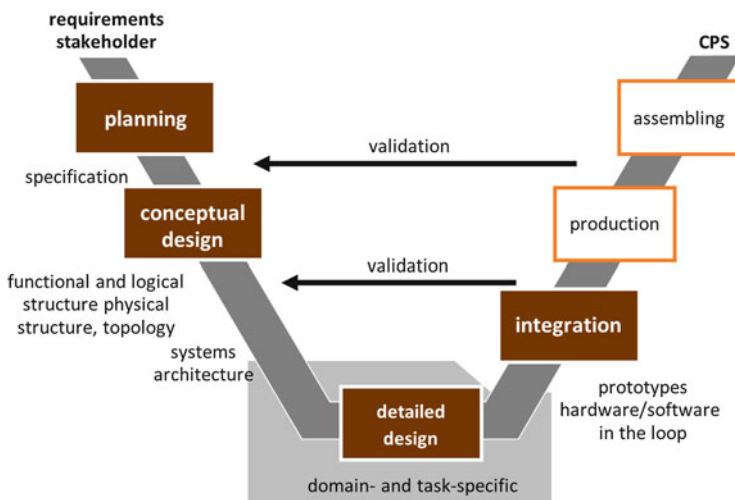


Fig. 2.8 V-model as a fundamental approach to CPS

In the *planning phase*, it is needed to analyse precisely the stakeholders of the CPS and their goals and expectations on the system behaviour. Since the CPS should adequately react to different situations, the definition of just one system response is not sufficient. Instead, it requires an intensive analysis of possible situations to specify the system behaviour. In addition, there remains a challenge, boundary conditions of the application environment, from the market and the competition, as well as from within the company to identify.

Characteristic for the *concept phase* is processing from rough to detail (Fig. 2.5). Under this phase it is necessary to substantiate the defined system behaviour by a functional analysis and to complement the logical order with function execution (operations). Structural analysis is the condition for a description of the physical architecture. In this context it is important to clarify how or what features are summarized in terms of subsystems or components. Hereby, considerations as the evolution of technical systems play a major role (Kossiakoff et al. 2010). Only in a few cases, radical new developments take place in product development. In general, it is necessary to consider incremental innovations. The physical architecture is then more or less given and serves as a basis for development. Product and variant management in the company are also influencing. This not only defines the product structure by the stored modularisation strategies. Finally, the physical architecture is characterized by the company's knowledge in development and by providing access to technologies. Linked to this is also the organizational assignment of development tasks to individual development teams in the specialized domains. Figure 2.9 summarizes these aspects of the system architecture together.

Result of the *design phase* is a system architecture, which describe structural description and system behaviour from different perspectives (Rechtlin and Maier 2000). The concept phase is of particular importance for development because it not only predefines the resulting CPS, but also establishes a system understanding,

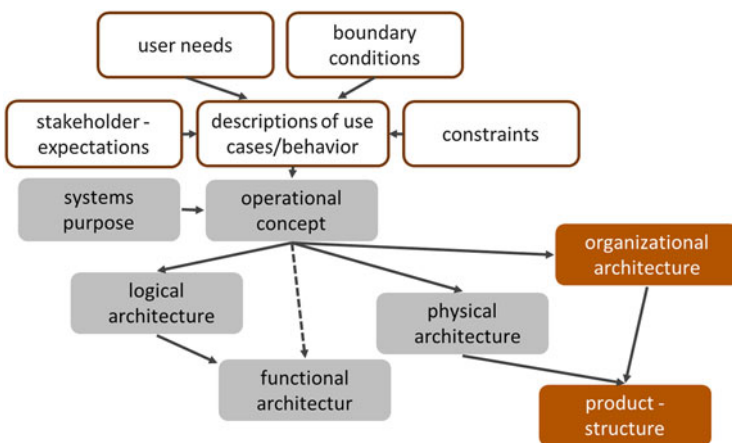


Fig. 2.9 Consideration to system architecture

to which the project partners must commit (Blanchard and Fabrycky 2012). Accordingly, the concept phase requires a highly interdisciplinary cooperation in which is assumed that the participants from different disciplines to develop a mutual understanding of the concerns of other specialized domains.

In contrast, the expertise and the specific knowledge of the individual domains are needed in the design phase. Since they summarize domain-specific knowledge, subject-specific process models are used to deal with the development tasks. In terms of the interdisciplinary nature of the entire development task, a special awareness of the overall system needs is necessary while the processing of subtasks. Decisions that were taken in one domain also influence the work and decisions of other development teams. A strong interface management is required, thus also in the domain-specific development phases, the overall objective won't get lost.

The phase of *system integration* provides companies especially in the development of CPS with special challenges. Individual solutions from different development teams, based on specifications, which only represents part of the complete the overall system requirements, are optimized in this way usually. It is in the nature of things that not all possible critical interfaces are considered to other subsystems. Often simply lack the domain-specific knowledge in detailed questions. In the phase of system integration, therefore much effort on the adaptation of individual solutions is put in terms of overall performance. This is associated with time and cost consuming iterations. It therefore makes sense to support the phase of system integration from the planning and design phase through methods and tools, as well as procedural and organizational interfaces by which interfaces between requirements, stakeholders, subsystems and functions between development teams and can be made visible. In addition, it requires methods to provide decision situations with sufficient data and information.

2.2.4 Systems Engineering as an Interdisciplinary Approach for Development of CPS

While in the previous chapter, the procedure from a more domain-specific point of view has been continuously broaden to draw conclusions for interdisciplinary product development, Systems Engineering provides an approach to support the theme from a cross-domain development perspective. Similar to the already shown systemic development approaches, Systems Engineering refers to a long history, (NASA 1995), especially for the development of large-scale systems (Chestnut 1967).

Systems Engineering (SE) is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining costumer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost, schedule, performance, training and support, test, manufacturing, and disposal. SE

considers both the business and the technical needs of all costumers with the goal of providing a quality product that meets the user needs. (INCOSE 2010, p. 6)

The definition shows that not only a focus on the stakeholders takes place, but also the entire product life cycle of the resulting system is made explicit in development. These limits of consideration are very broad in terms of description of a socio-technical system from the outset.

Also as part of SE approaches, development of technical systems is understood as top-down process with iterative character (Eisner 2008). The focus lies more on a total system approach and on how to find an optimal solution for complex tasks and problems (Hitchins 2007). Of course, in SE detailing phases in the domain-specific development are also needed. Here is less discussed how to proceed in detail, but more which tasks a system engineer has to fulfil, to be able to reasonably coordinate the results of these domain-specific phases to each other and to be able to integrate them into an overall system. The methods used in the context of the SE therefore primarily focus on the control and management of complexity (Kossiakoff et al. 2010), analysis and description of interactions between to developing systems and the environment as super-system, and between the sub-systems. Especially the latter is of course based on the expertise of different domains, but should be supported by the system engineer, from whom a broad technical understanding is expected (Blanchard and Fabrycky 2012).

Oliver et al. (1997) explain that SE approaches need to support two ways of looking at development:

SE management process describes the technical and organizational effort within the product lifecycle and thus focuses more on the typical tasks of project management. Target of management procedures is to provide information and to evaluate it in order to support the decision-making in terms of trade-offs between efficiency and costs (NASA 1995). It is necessary to consider restrictions on cost, time and potential risks, both in the development itself as well as in the evaluation of the performance of the technical system.

SE technical process includes all activities from the first request of requirements up to development through to verification and validation of the results. Both, procedures represented in literature as well as procedures known from practice, based on systemic approaches and mind sets that have been explained in detail in the previous chapter. As process model now commonly a V-model is used as basis (e.g. (Blanchard and Fabrycky 2012; Kossiakoff et al. 2010)).

Typical methods of SE focus less on the support of individual activities but more on the analysis and description of the system complexity. Therefore, it should be given particular mention to:

- Methods for modelling and simulation of complex systems (e.g. NASA 1995)
- Methods for analysing system contexts both at a functional and structural level; for this, graph-based approaches are used as well as network-based approaches (conclusion e.g. in (Parraguez 2015))

- Methods for the preparation and presentation of data and information flows in the technical system, taking into account constraints and stakeholders through approaches of model-based systems engineering (MBSE) (e.g. Delligatti 2013).

From the perspective of systems engineering, the emphasis in the description of data and information flows lies in summarizing the variety of data and information and in preparing a structure to firstly identify interfaces between sub-systems and the environment within the technical system. To this end, data and information from the synthesis of partial elements are grouped in such a way that dependencies between elements of the overall system can be identified.

Especially the extensions within the V-model for development of interdisciplinary products such as CPS give an indication what to look for in design but for the concrete process design and support, they are only partially suitable. Here, a significantly more granular fine-mapping of processes is required. Therefore, in the following chapter approaches are presented to substantiate and refine the process descriptions.

2.3 Concretisation of Process Descriptions in the Sense of a Workflow

Especially the extensions within the V-model for development of interdisciplinary products such as CPS give an indication what to look for in design but for the concrete process design and support, they are only partially suitable. Here, a significantly more granular fine-mapping of processes is required. Therefore, in the following chapter describes approaches to substantiate and refine the process descriptions.

Looking at the data and information flows within the context of process models, both, product models as a result of synthesis and analysis models shall be considered. This requires a detailed process description, which in turn forms the basis for the design of IT structures. It should focus on that system descriptions within the process are successively refined through the use of different methods and related tools. This leads not only to different versions within a product model but also contributes to a plurality of partly also very heterogeneous product models, which need to be managed and tackled. With increasing maturity of the technical system, the data quality improves and data uncertainties are gradually reduced.

Analysis models are based on product models and the associated level of maturity of the considered subsystem. In addition, analysis models depend in their detailing or their structure from the objectives which are pursued with the analysis in each case. Product models require a corresponding contextual preparation to make them accessible for the analysis.

Both model types, product models and analytical models contain data and information which describe each viewpoints or parts of the resulting CPS. For preparation of the product models for analysis and specifically for simulations,

engineering-workbenches find use. The aim of this is, on the one hand to support the modelling for simulation by the prepared context information for pre-processing and assigned to the product model. On the other hand, the analysis results are pro-cessed in such a way that they are within the meaning of the decision support for the development process available in post-processing. Libraries for modelling, load cases, material values, etc. supplement this context information for analysis and can simultaneously be understood as knowledge repositories.

For collection and delivery of all product artefacts various IT tools are available. Product data mainly from construction will be deposited directly into product data management systems (PDM systems). Depending on the structure and format of these tools, this data can be linked to analysis results. Such processing and provision of data and information facilitates the provision of data for individual development steps and the development situation significantly, but also implies that the PDM systems need to be adjusted in terms of product lifecycle management to the development processes. In addition to PDM solutions for managing the documents, data filing systems in which the files themselves are stored, are used. This in turn is caused by the use of various tools in development. In sum it is necessary to support development processes with a diverse IT landscape. Their efficiency depends on how successfully typical development processes for the own organization can be mapped and finally adapted to the data and information flows.

Figure 2.10 gives an indication to the components of a holistic IT infrastructure for development. Besides PDM tools and engineering workbenches support file storage systems the structuring and management of data. These are linked to the for development necessary producer systems both for the synthesis (for example, CAD) as well as for the analysis (for example, FEA, MBS). Within these IT systems in turn, templates, libraries and assistance systems find use to support individual

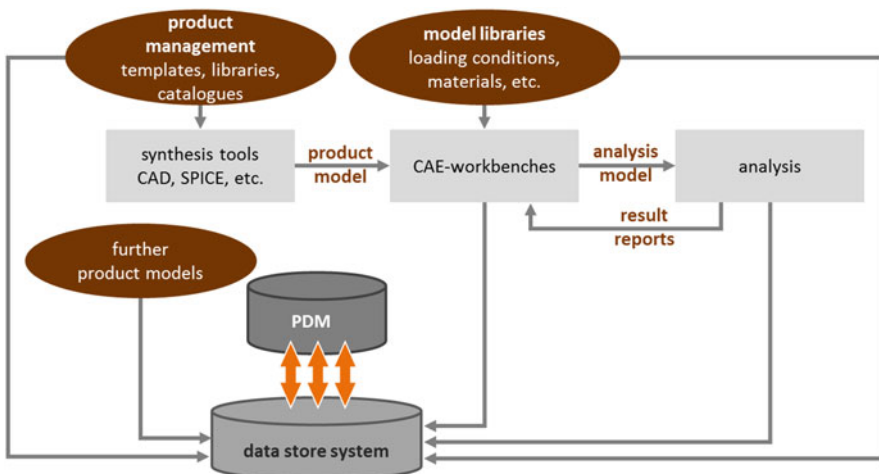


Fig. 2.10 Overview of components of an IT infrastructure for development

tasks through targeted provision of knowledge, but also to enable a holistic and overarching product management. Finally, it requires suitable data interfaces to ensure the exchange and further use of data within the development process. It quickly becomes clear that for a concrete definition of an IT infrastructure, a detailed treatment of the data and information flows is necessary, which cannot be guaranteed by the process models described in the previous chapter. The process models require a significant specification, for which the influencing factors are described below.

2.3.1 Adaptation of Development Processes to the Context

The refinement of the development process is not only required to substantiate to processes in the development and to support decision-making processes through targeted information delivery. This is done not only on the basis of interdisciplinary tasks in the development but also by the integration of the development in the company's organization. The link to production and assembly, sales and marketing, logistics, procurement etc. is especially required to consider all influences on development in terms of product lifecycle reasoning. As a result, so called swim lane models for process description can be found in the companies (exemplarily shown in Fig. 2.11). This also goes hand in hand, that for development necessary IT structures are linked to the IT tools of other company divisions. In this context should be referred specifically to the link to Enterprise Resource Planning systems (ERP systems).

The processes depicted with such Swim lane models are very company specific. They resemble each other but generally are not equal because they are influenced not only by industry, competition and market but also by corporate strategies. Here, for example, aspects such as competitive and technology strategy or typical customer patterns play a decisive role. Such process descriptions and therefore the support of developers are not only based on typical processes, but also can be explained to some extent from the company's history. Last but not least are manifested in the development processes and the associated IT structures the experience and knowledge of the company.

Because of the diversity in the expression of the factors influencing the process design, a generic oriented refinement makes little sense. Below shall therefore be briefly outlined, which factors are in what way to take into account in order to reflect the company's internal processes as a whole can. In turn, starting points and specifications for IT structures can be derived.

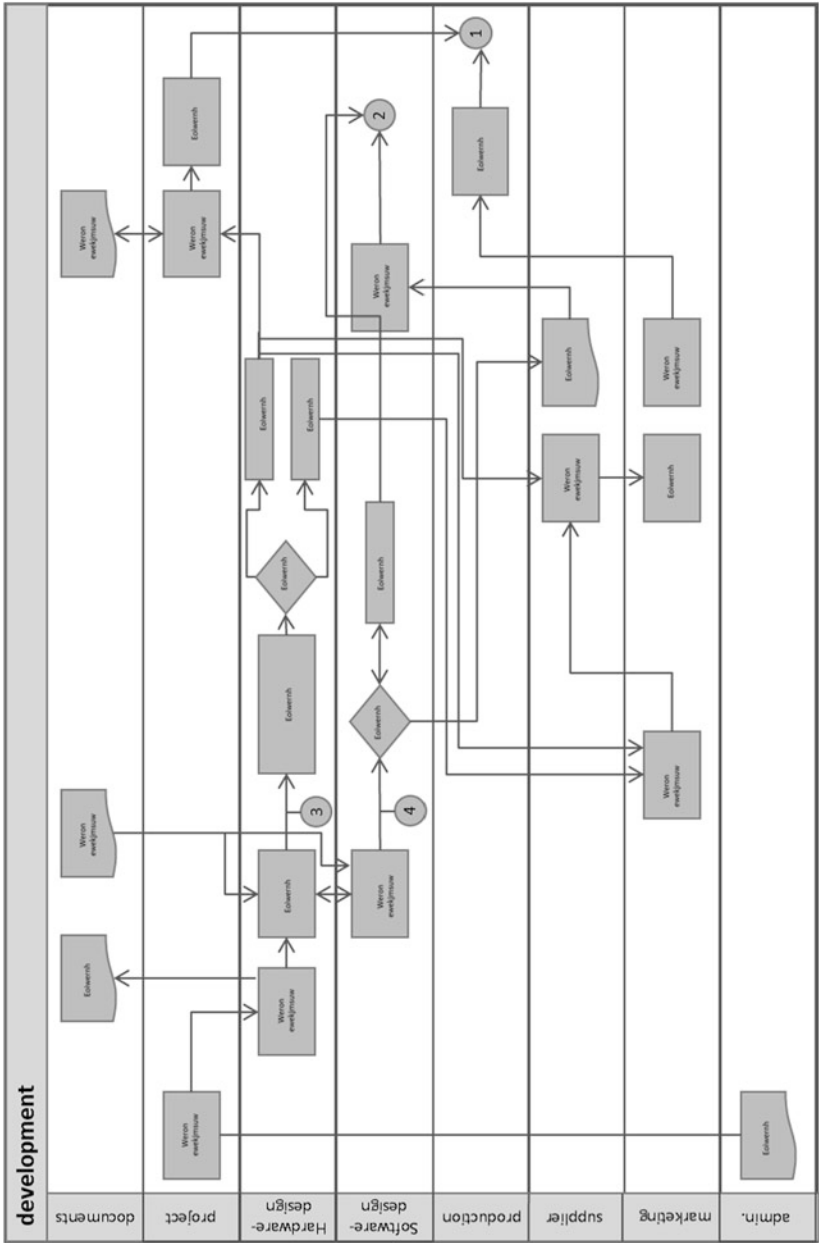


Fig. 2.11 Example of a development process as swim lane model in the company

2.3.2 Identification of Context Factors for Adaptation of Development Processes

The adaptation of the generic process models is described in the literature for two reasons: on the one hand requires the pronounced interdisciplinarity especially for CPS the consideration of typical procedures of disciplines involved to obtain functional and high-quality solutions for components and subsystems (e.g. Pugh 1991; Gericke and Blessing 2011). On the other hand, the adjustment of development processes to the corporate context (industry, market, product group strategies) is necessary to explicitly include the resulting specific framework for development (e.g. Skalak et al. 1997; Meißner and Blessing 2004). Gericke et al. (2013) illustrate two types of customization options of development processes: the Augmenting and Tailoring. The Augmenting ultimately describes the extension of the procedure description not only by additional process steps but also by the integration of additional information such as guidelines, design practices, specific methods, or the like. Target of Tailoring is to carry out industry or company-specific adaptations of the development process, whereby it is mainly about to make the process steps explicit, which results from the product or the industry (Gericke and Moser 2012). In the aviation sector this includes, for example, admission procedures or hedging measures. For CPS are, for example, specific measures in relation to consider the data security or the human-machine interaction, required. The challenge for both forms of process adjustment is to completely grasp the development context and describe it in its characteristics. Hales and Gooch (2004) illustrate this in a framework to classify those factors influencing the product development and to identify (Fig. 2.12). Gericke et al. (2013) use this structuring approach and assign more detailed context factors that have been identified based on a comprehensive literature review.

The resulting list of over 230 contextual factors (in Gericke et al. 2013) provides basic evidence, what to look for in the concrete detailing, but for practical

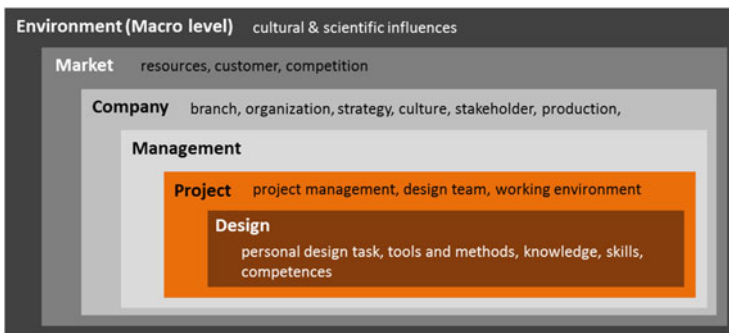


Fig. 2.12 Influencing factors to product development according to (Hales and Gooch 2004) improved in (Gericke et al. 2013)

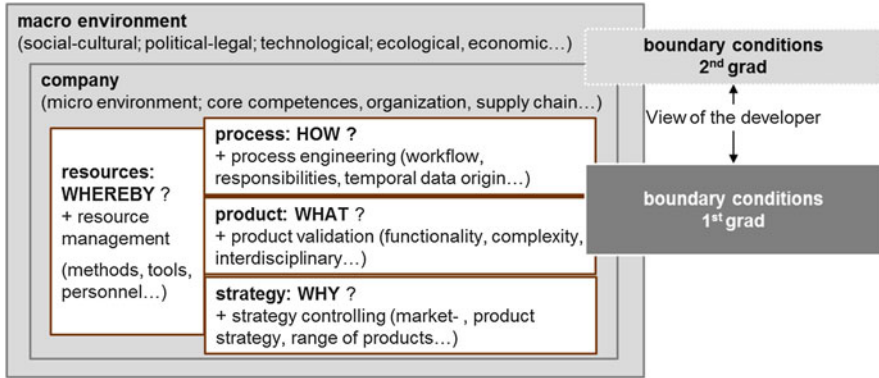


Fig. 2.13 Detection of the contextual factors for describing the data and information flows (Reitmeier 2015)

application, this list is a bit bulky. Also not included in the rather linear view is the fact, that between the contextual factors a number of dependencies are observed.

Reitmeier proposes to extend the structuring approach (Reitmeier 2015) to capture the data and information flows by continuously evaluating the contextual factors with respect to the decision to be made (Fig. 2.13).

In (Reitmeier 2015) is distinguished for the classification of contextual factors in such of first degree, which are directly connected with the activity of the developer and thus also directly can affect the necessary activities, or even can be influenced by the developer. In addition to the aspects of the product and the process, the company's resources also play a role here. Not least, strategic considerations such as the importance of the development project for market, industry, customer or integration in the product management influence a role. These factors influence the adaptation of the development process to the development task. Boundary conditions of second order, which at a higher level can be derived from the macro environment or the company's strategy, however affect the basic design of the processes in the development.

2.3.3 An Approach for Systematic Analysis of Determining Factors for the Development Process

In addition to detection and description of the contextual factors it requires an analysis, which indications can be derived for the design of development processes. For this, a systemic approach after (Negele 1998), which in turn follows an idea of (Patzak 1982), is used. This approach of Negele will be introduced in the following chapter.

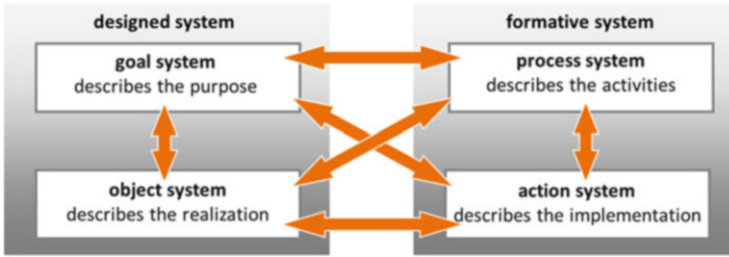


Fig. 2.14 The development of CPS using the ZOPH model after (Negele 1998)

In principle, it is distinguished between the system to be designed, here the CPS as product, and the formative system, that is the process of targeted activities that lead to this CPS. Both systems, the CPS and the development process cannot be considered independently. Considering further that the CPS initially is present only as an aggregation of targets, which are substantiated by development, it follows that the CPS is again divided into a target system and an object system. This opens up the possibility to address the uncertainties in development. On the other hand, in connection with the process is illustrated the extent to which the resulting CPS is correlated with the objectives. A similar approach is for the process to derive even. The formative system of the process is mainly determined only from the logical approach and has a generic character for the product group. The concrete development task can be derived only if demands are accurately specified, resources and responsibilities are defined and a termination occurs. For this reason, a specific project will be derived on the basis of the processes virtually, resulting in the distinction between a process and a system of action.

Finally, of course, it requires a definition of the system boundaries, which are characterized not least by contextual factors on the micro and macro level. The difficulty here is that this system boundaries often cannot be drawn strong and clear, because it is precisely the characteristic of the CPS that they interact closely with the environment or other systems in the environment. CPS must therefore be interpreted frequently in the context of a system of systems (SoS). Such blurred dividing line is, however, difficult to tackle in development. Therefore it is necessary to give special care on the delimitation and description associated in- and outgoing data, energy and material flows. Figure 2.14 illustrates this model approach after (Negele 1998). The name *ZOPH* of the approach results from the first letter of the german words of the partial models (**Z**iel, **O**bjekt, **P**rozess, **H**andlung).

This raises the question of how this mind set supports the process development and control of the development and the associated organization of data and information flows. For this purpose, the individual subsystems shall be further concretised in reference to (Negele 1998).

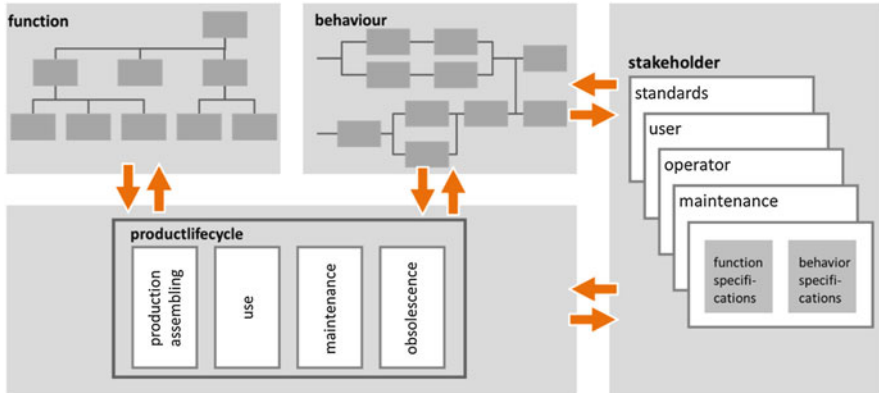


Fig. 2.15 Detailing of the target system

2.3.3.1 Goal System

The goal system classically summarizes all the requirements for the evolving CPS. As a product from the capital goods sector, here one needs to distinguish between load and performance specifications in principle. The challenge lies in the completeness of goal acquisition. Here are not only partial functions and their expression of importance, but also the description of the characteristics of the desired operations. For structuring the target system approaches for describing the system architecture can be used, which reflect both functional and structural aspects. Since generally in development always similar systems are in focus, may be used for the CPS also established product structures. Relationships between functioning synthesis and principle synthesis (Andreasen and Hein 1987) need to be considered especially for completely new developments. For adjustments, variant developments or incremental developments, these aspects are already included in the system architecture.

The complexity of the CPS results not only from the functional diversity but also from the diversity of the expected behaviour, which is not least determined by different stakeholders on the CPS to be developed. Therefore, the target system shall additionally be analysed with respect to different types and forms of operations, what also integrates aspects of the product life cycle at the same time. This requires a detailed analysis of the stakeholders, as summarized in Fig. 2.15.

The goal system includes aspects of system definition, therefore the answer to the question of which functions can be realized only through the CPS and which arise from the interaction with other systems. One important result of the analysis to the target system must be to detect cross connections between functions, product structure, operations and stakeholders to present and show mainly conflicts between these. Ultimately, the target system is a reference for the object system, but also for the process system.

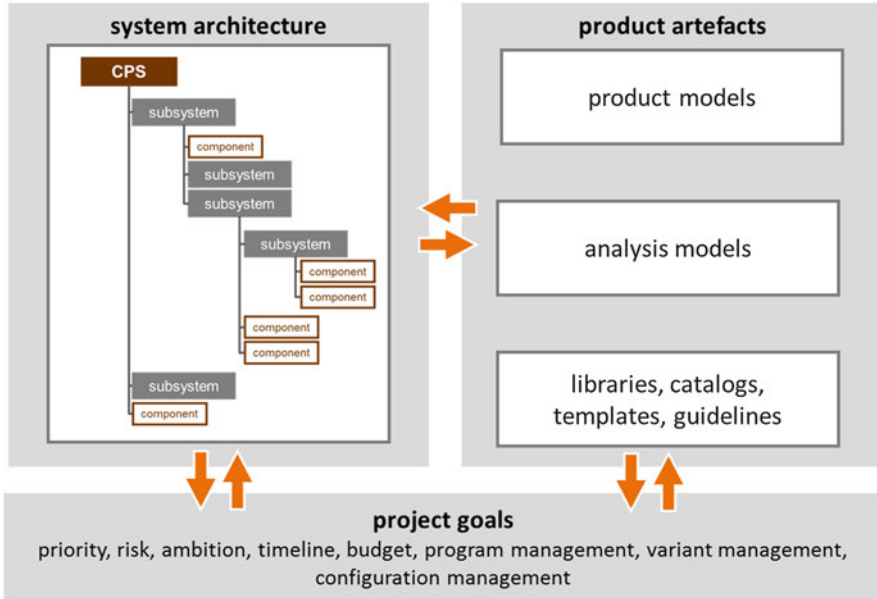


Fig. 2.16 Specification of the object system

The target system is manifested in the requirement description, which in turn is the basis for requirements engineering. In conjunction with the stakeholder description and hedging measures, risk assessments or reliability statements can be deduced.

2.3.3.2 Object System

The object system ultimately summarizes all results in the development process and therefore also all product artefacts in the various phases of the development process. This includes any form of additional information such as libraries, catalogues or templates that underlie the development in the different producing systems (Fig. 2.16). Ideally, the structuring is based on the product architecture, which generally underlies also the target system description. A consistent structuring of both, the object and the target system is the basis of the balance between demands and results of the development process. However, the orientation on product architecture takes generally into account the participating disciplines for component development.

The product artefacts that are collected in the object system are available, both as documents in data storage systems as well as hardware (prototypes, hardware-in-the-loop). The analysis of the object system therefore not only serves to flesh out the IT structures but also the identification of data-technical necessary interfaces

between the product artefacts. The relationship between various product artefacts provide, at best, product data management systems and engineering workbenches. The relationship between data and information from other business sectors are ideally represented by ERP systems. The enrichment of product artefacts with information like date of creation, creator, etc. Versioning provides the one hand the connection to the system of action, on the other hand, these also can be assigned to phases within the development process and thus draw conclusions for the degree of maturity.

2.3.3.3 Process System

The process system defines in detail the activities and actions which are relevant for the specific development. Logical dependencies result from the physical relationships in the specialized domains as well as from the to-implement system functions and the expected operations. The structure of the process system should be such that the analysis of data and information flows can occur between the individual acts. The individual process steps are considered quasi as transformations of the initial state into a desired end state. For this transformation are usually available the experience and knowledge to the system, which provides guidelines and methods that take into account not only the physical relationships but also best practices. Both, methods and best practices include ultimately guidelines for action and thus provide the detailed procedures a framework for action. For the analysis of process steps therefore the proposed model Fig. 2.17 with reference to (Birkhofer et al. 2005) can be used.

A generic procedure description also results from the problem-solving cycle (see Sect. 2.2.1), because there with a classification of methods and the data characteristic is connected. As described above, the problem solving cycle is ultimately to go through at every stage of development. This results is not only a classification approach for methods (see Fig. 2.3) but also a statement of the maturity of the resulting system. Conversely, this means that each method must be adapted to the maturity level of the product. With the application of methods in practise usually IT-tools are connected, which data requirements and their evaluation results are directly attached to the methods or the best practices. For the description of the development processes, the development influencing tasks like procurement, purchasing or production and assembly now also have to be considered, as they either provide input or need input at a defined time. Via the process system one can recognize and specify procedural interfaces. The process system is manifested in the swim lane models mentioned above (Fig. 2.11).

2.3.3.4 Action System

The action system describes the organizational framework or the organizational impact parameters for the process design. The organizational structure results in

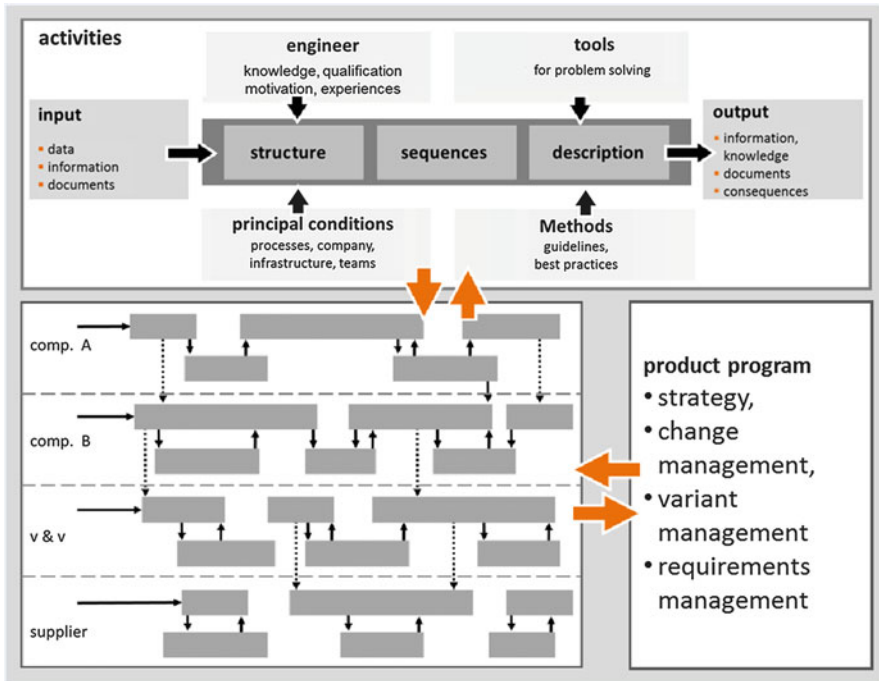


Fig. 2.17 Specification of the process system

responsibilities, role assignments and responsibilities for partial activities for both, the actual development work as well as to administrative supporting processes such as release mechanisms etc. In action system also material and personnel resources are defined, whereby the material resources available especially IT tools, licenses and their characteristics are included, but also test stands, laboratory equipment, test equipment, etc. are covered. Since today a variety of development services is adopted by service providers or suppliers, these have to be interpreted as parts of the action system, which supports the development with resources and capacity.

The action system serves in addition to content issues also to scheduling specifications of development processes, depending on the development task and possible priorities of the company. Depending on the development task, deadlines and responsibilities must be assigned to the individual process steps. For the purpose of a multi-project management, this assignment must be accompanied by a project prioritization, which results from the program or product management. Thus, directly from within the action system, the processes are complemented by typical information to substantiate the workflow in terms of project management. The components of the action system are summarized in Fig. 2.18.

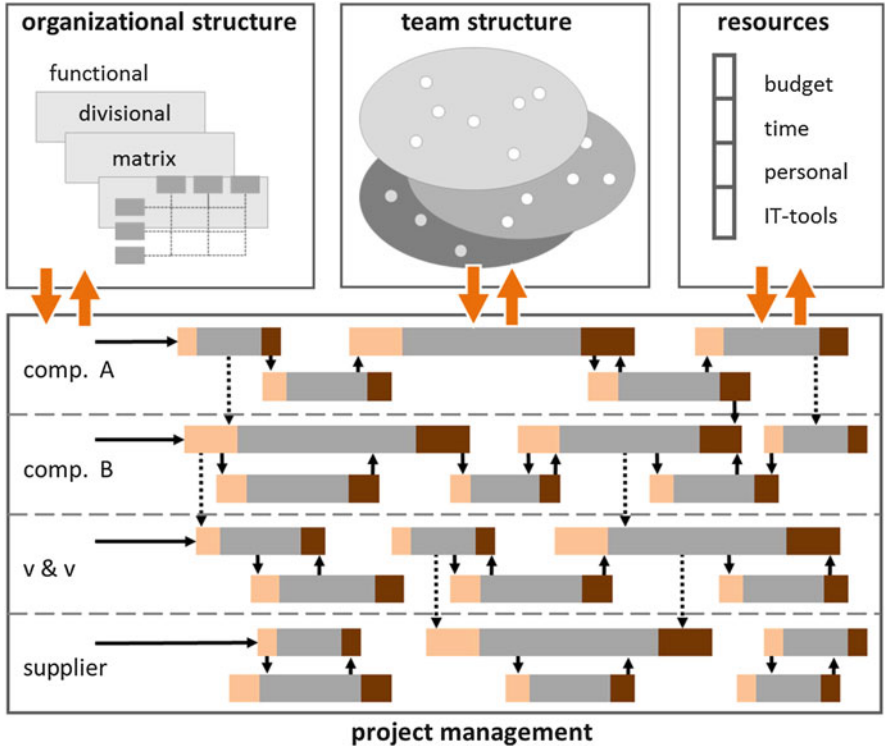


Fig. 2.18 Specification of the action system

2.3.3.5 Control of Development Tasks via the ZOPH Approach

The individual subsystems of the ZOPH model are not independent, as is already explained in Fig. 2.14. In terms of a holistic process management it is rather necessary to consider and develop all the subsystems equally. However, various views on the development process can be mapped over the subdivision into subsystems. The planning and integration of IT structures for integrated data and information flows, for example, requires not only the consideration of the typical system architectures but also of typical processes to tightly coordinate the individual IT tools. For supporting decision-making processes in development in turn the provision of all available data is required, but it also requires the knowledge of organizational, process and product-related interfaces that are clearly shown by the ZOPH approach. This is the basis on which to assess the impact of decisions and to inform those who are affected by the decision.

The main objective of the development is undoubtedly the CPS. This is also connected to a number of side objectives, such as quality, reliability etc. For this purpose usually consuming associated processes such as risk management or requirements management are implemented in product development, which are

often to be considered as parallel to the actual product development process. Configuration management and change management on the other hand consider aspects of creating variants, which are normally anchored in program management. These associated processes, for example, determine the product development via modularization approaches, common parts and repeat parts concepts etc. With the analysis of the company within the meaning of the ZOPH approach, these processes can be mapped in a holistic way and coupled with development. Ultimately, these associated processes grab, even if they have different origins, back to specific elements in the subsystems and bring them goal-oriented together.

2.4 Model-Based Engineering for Mastering Complexity

The major challenge in mastering a CPS, nowadays lies in the system integration. For individual activities or aspects in development now a variety of powerful methods and tools are available, the overall system approach is, on the other hand, less pronounced supported.

Especially in the data and information flows are always media discontinuities to tackle, which easily lead to fractures in the reality in development. There are a variety of documents such as request lists, CAD models, feasibility analyses, calculations as well as Power Point or Word files generated which describe the resulting product both on functional as well as on structural plane. These are for individual development steps or specific decisions of importance, but are available in very different formats. In addition, they represent some very different perspectives on the CPS, which makes their interpretation and exploitation difficult. The cause of the resulting lack of development lies accordingly in the document-based management of the data and information flows, which neglect the data characteristic. The preparation of more or less unstructured data requires special effort, for which the creation of data exchange formats is often insufficient. An illustrated document-centric view in general supports only specific views on the CPS. The challenge arises not only from the different data formats and data structures but also from the fact that with the various activities (synthesis/analysis) different data characteristics are connected (properties/features), which complicates comparability and stringent further usage.

To overcome these development disruptions, and thus to ensure a certain completeness and efficiency of the development processes, increasingly model-based approaches are tried. With such a model-centric organization of data and information flows, the model becomes the source of relevant information by structuring the information according to a predefined scheme (Weilkins 2008).

Model-Based Engineering (MBE) should here be understood as an engineering approach, which uses models within the meaning of pre-structured data as a basis for the data and information flow. These models include all this data and information relating to the product life cycle, based on the requirements of the

design, verification and validation of subsystems as well as on the overall system (NDIA 2010).

Model-based engineering may be understood as an overriding principle that originally sprung up in the development of software-intensive applications. The application of CPS is not only beneficial to master the complexity in development but also due to the system characteristic. The basic idea here is to continuously hedge the system behaviour at different levels throughout the entire development process by the use of these model approaches. For this purpose, firstly parameter-based models (for example, in Matlab Simulink) are constructed from the overall system, by which the system response can be displayed and analysed. These are successively refined, both, in terms of modelling approaches as well as by hardware components that are integrated (Albers et al. 2013). This ensures the system integration from the early stages of the development.

Meanwhile, the understanding of MBE is somewhat broader and also includes in many cases Model Based Systems Engineering (MBSE) and Business Process Modelling (BPM) (Lee 2008).

Difficulties or challenges in the dealing with models arise from their properties (Stachowiak 1973). Thus, models are only depictions of a system. This will entail contractions: by means of the attributes from the original, only those things are recorded, which are relevant for the defined viewing purpose. By assuming a certain substitution function, therefore models give only a specific limited view to the original again (Stachowiak 1973).

A further distinction needs to be made for models:

- Especially in engineering, the feasibility of models is presumed. Models are thus constructed under the use of method tools such that they can be used as basis for simulations. This opens the possibility to derive statements about behaviour, performance and function as early as the early stages of development.
- Increasingly descriptive models that allow a symbolic representation of the CPS with a defined syntax and semantic, gain in significance. These models are information models, in which feasibility generally does not exist (exceptions can be found in software development, where such a source code can be derived, see also (Rumbaugh et al. 1999)). In this context, UML or SysML models are exemplarily mentioned. In the end, even CAD-models, which initially support structure visualization and are based on a pure geometry-based view, are included. Simulations cannot be initialized directly but always the preparation of data for an analytical model is required.

The two different interpretations of the MBE are summarized once again in Fig. 2.19. This depiction illustrates that the model-based approaches on the basis of executable models correspond to a vertical integration. Accordingly, the challenge is to refine the models depending on the progress of development, or to link the different modelling approaches to make them accessible for simulations (Paetzold and Reitmeier 2010). MBSE other hand focuses on horizontal integration. The goal is here to compile all data and information for the CPS over the product lifecycle

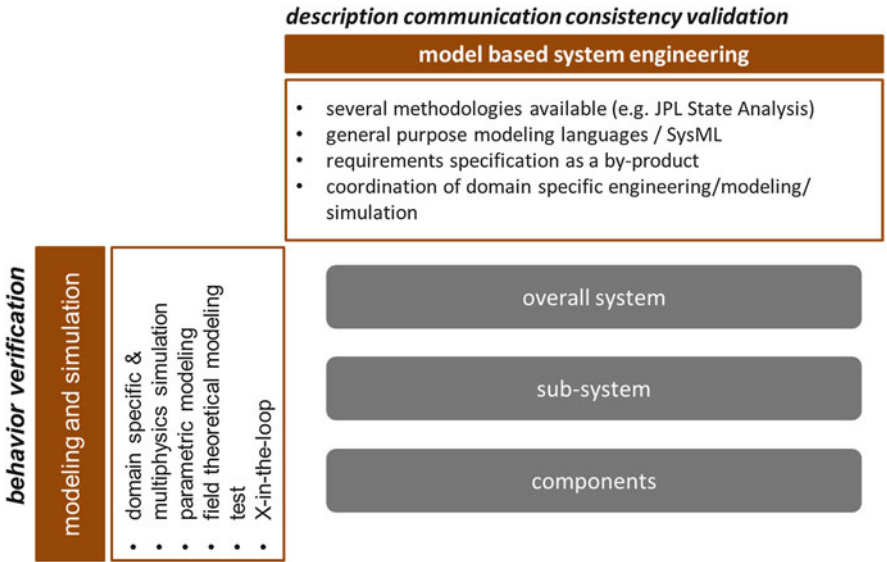


Fig. 2.19 Different views of the model-based engineering (Omiciuolo et al. 2015)

in order to ensure a holistic view on the resulting technical system or to assemble different views.

With vertical integration, especially the kind of modelling or the choice of the modelling paradigm is combined. From a systemic point of view one can distinguish both, on the structure level as well as on the function level or the behavioural level. A modelling based on the structure definition for the whole system is extremely difficult, what is caused by the domain-specific physical and related mathematical descriptions (geometry, source code, integrated circuits).

For the simulation of the system behaviour it is more likely to resort to parameter-based approaches to modelling, as these are based on the more generic functional description and are linked via input-output relations. With this, although one accepts a higher level of the abstraction model, however, succeeds a cross-domain modelling, which can be understood and interpreted in the domains. The challenges that are associated to this observation will not be further discussed here, but reference is made to additional literature (the problem of universal description language, for example (Panreck 2002); multi-domain vs. domain specific, for example (Sangiovanni-Vincentelli et al. 2009).

2.4.1 Model Based Systems Engineering

The second aspect mentioned in the context of model-based engineering is also treated under the concept of model-based systems engineering (MBSE).

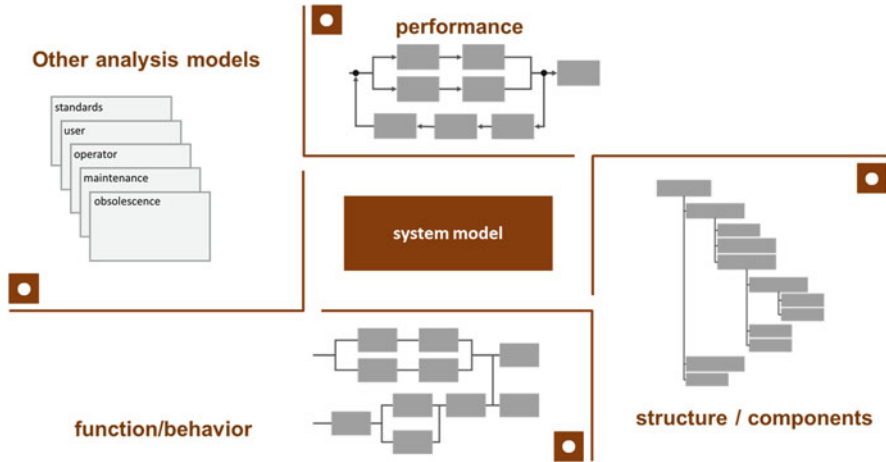


Fig. 2.20 Perspectives which shall support the system model

Model-based systems engineering (MBSE) is the formalized application of modelling to support system requirements, design analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. (INCOSE 2007)

The basic idea is to support the development process and especially the data and information flows through system models that depict not only aspects of the entire product life cycle but also the perspectives of different stakeholders. This is intended not only to support development processes but also to provoke balanced system solutions (Eigner et al. 2012). For this purpose, as a basis a unified modelling approach is used, which allows to map both, system models as well as development activities. MBSE thus should be understood as methodology; that means a summary of processes, methods and tools by which a defined goal, here the integral development of complex technical systems, is announced (Estefan 2007). The management of complex technical systems is supported by the fact that not only different perspectives on a CPS can be distinguished but also between function, structure, behaviour and performance. The possibility of hierarchy formation facilitates impact analyses or traceability of design changes (Fig. 2.20).

In order to deploy and support a methodology efficiently, three aspects must ultimately be integrated (Delligatti 2013):

- Methods which describe the development of the technical system,
- Languages that define grammar as a set of rules for system description and that are understood by all parties,
- Tools that translate the languages and that support in construction and interpretation of the models by providing a development environment and routines.

These aspects will be described briefly below.

Methods within the meaning of MBSE describe action rules for consistent configuration of system models. These methods are not only influenced by the purpose of the model but also by the mind sets and ways of abstraction of those who wish to use these models. Of course, the methods follow very much the logical and systemic approaches for development and the structuring of complex technical systems, as they are described in this chapter. Certainly, here play for example domain- or industry-specific priorities in the concretization of the methods such as for example domain-specific development and lifecycle models a role. Also specific aspects of the system Engineering management process, such as risk management issues or security issues, may affect the method in detail here. These methods reflect quasi a type of a modelling philosophy.

For MBSE, in the literature a number of methods can be identified. In the context of Systems Engineering as an integrated approach to development by CPS, one should particularly mention:

- OOSEM (Object-oriented Systems Engineering Method), which was designed as a top-down approach by the INCOSE, based on a functional analysis (Friedenthal 2014).
- SYSMOD method, a standard top-down approach developed by Tim Weilkens, which is based on UML methods and expand them in the sense of systemic product development (Weilkens 2008).

Besides that, a number of other methods, such as IBM Telelogic Harmony SE, IBM Rational Unified Process for Systems Engineering (RUP-SE), JPL State Analysis (SA), Dori Object Process Methodology (OPM) exist. A detailed summary and explanation of these methods can be found in (Estefan 2007).

Modelling languages define the grammar, thus the nature of the elements and their connections by means of which a model must be built to represent defined contents intelligible. The language that is spoken, directly influences the way how the technical system is seen. The dilemma of the description language for interdisciplinary technical systems has long been known (for example, Panreck 2002). Domain-specific languages are generally not suitable to transmit content beyond the domain boundaries or to integrate domain-external matters into the model. Again and again, attempts have been made to develop general description languages, which on the one hand reflect the system holistically and are intelligible to another. This dilemma can also be seen for languages for MBSE. Exemplarily, reference is made to work, which indeed succeed in depicting geometry elements by using SysML (Eigner et al. 2014), but these are not readily to interpret as intuitive as a CAD model. They therefore require to “oneself empathize” into the language, to understand these.

For MBSE primary graphic-oriented languages are used, which not only define the element types which may be used in the model but also the relationships that are allowed between the element types. This is complemented by a notation which allows the display of elements and relations in diagrams. In the past, each of the developed method has often developed their own language, what leads to a corresponding diversity. In this context, *System Modeling Language (SysML)*,

Unified Modeling Language (UML) and *Integration Definition for Functional Modeling (IDEF_x)* are particularly mentioned.

All graphic languages have in common that charts are available for both, the structural view and the behavioural representation by which in the systemic sense both, structural and behavioural models are constructed and interrelated. The specific designations or aspects of hierarchy and points of view can, however, vary.

The tools for MBSE implement ultimately the language and modelling methods. They support the engineer by the availability of a graphical interface, through routines and libraries for creating and linking models and the automation to update them. Commercial importance have *Cameo System Modeling/MagicDraw* (vendor: NoMagic), *Enterprise Architect* (vendor: Sparx Systems), *Rhapsody* (vendor: IBM), *Artisan Studio* (vendor: Atego) or *UModel* (vendor: Altova).

The benefits of using MBSE-approaches for interdisciplinary development are obvious. By provision of clear and accurate models can also interdisciplinary aspects be summarized in a model approach. The emerging information models are undoubtedly suitable to reflect and to summarize different perspectives on the CPS in different hierarchical levels. This also results in the ability to detect dependencies from both, a procedural, product-oriented as well as an organizational point of view and to derive interfaces more precise.

Because of these advantages is seen in both, academic as well as industrial environment, a considerable potential for development support by means of MBSE. Nevertheless, MBSE approaches are not yet established very wide. Challenges arise not only for the utilisation of the resulting information models and their integration into the data and information flows respectively the IT infra-structure but also in the creation of these models.

The models which are created on the basis of MBSE are not directly available for simulations. Although today parameter-based models which then e.g. are carried out by Matlab/Simulink (Sop Njindam 2015) can be derived from certain types of charts (block diagrams) commercial data interfaces are currently not yet provided for this. Now, there are some interesting approaches to integrate the MBSE-approaches into the IT structures of companies, for example, the coupling of requirements management with MBSE-approaches (e.g. Eigner et al. 2015a, b) with the objective of repeat and further use of the data and information.

For a successful integration of MBSE models into the data and information flows, and thus also in the IT structures of a company it is necessary that these reflect the product and process structures. This in turn presupposes that the development methods must be refined to such extent that typical development challenges and circumstances are addressed. The methods mentioned are, similarly as shown for the process models in this chapter, still relatively generic. For an effective use it is necessary to substantiate them and to adapt based on the specific development conditions. This not only requires an intensive analysis of the structures of the CPS but also of the development processes, which are reflected in the MBSE modelling methods and the models themselves.

In this context, also the question of the modelling depth for the technical systems which should be imaged is posed. In the literature currently the trend, to use MBSE

approaches down to very low levels of detail, can be observed (e.g. Eigner et al. 2015a). However, the modelling depth depends not least on the model's purpose, especially if one is considering that the derivation of executable models out of MBSE models is associated with considerable effort. In addition, in each domain very powerful modelling and simulation tools are available, which on the one hand can depict the subject-specific contexts very well and in detail, and on the other, can be easily understood and interpreted by experts.

The strengths of MBSE lie in the possibility of an overall system view and the associated possibility to recognize interrelationships and interdependencies in the CPS and provide them for development processes. This aspect should therefore define the modelling depth for MBSE models.

References

- Albers, A., Behrendt, M., Schroeter, J., Ott, S., Klingler, S.: X-in-the-loop: a framework for supporting central engineering activities and contracting complexity in product engineering processes. In: ICED 2013: 13th international conference on engineering design, Seoul, South-Korea (2013)
- Andreasen, M.M.: Machine design methods based on a systematic approach. Dissertation, Lund University, Sweden (1980)
- Andreasen, M.M., Hein, L.: Integrated Product Development. IFS, Bedford (1987)
- Birkhofer, H., Jansch, J., Kloberdanz, H.: An extensive and detailed view of the application of design methods and methodology in industry. In: Samuel, A., Lew-is, W. (Hrsg.) Proceedings of the ICED 2005, Melbourne (2005)
- Blanchard, S.B., Fabrycky, J.W.: Systems Engineering and Analysis. Verlag Pearson New International Edition (2012)
- Boehm, B.: Guidelines für Verifying and Validation Software Requirements and Design Specifications. In: Samet, P.A. (ed.) Euro IFIP 79. North-Holland Publishing Company, Amsterdam (1979)
- Boehm, B.W.: A spiral model of software development and enhancement. *Computer* **21**, 61–72 (May 1988)
- Chestnut, H.: Systems Engineering Methods. Wiley, New York (1967)
- Delligatti, L.: SysML Distilled. Addison-Wesley (2013)
- Ehrlenspiel, K., Meerkamm, H.: Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit. 5. Auflage. Carl Hanser, München (2013)
- Eigner, M., Dickkopf, T., Schulte, T., Schneider, M.: mecPro² – Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML. In: Schulze, S., Muggeo, C. (eds.) Tag des Systems Engineering, S. 163–172. Hanser Verlag, München (2015a)
- Eigner, M., Eickhoff, T., Ernst, J., Eiden, A.: Systemübergreifendes Änderungsmanagement zwischen PLM und ERP. In: PLM Jahrbuch 2016 – Ein Leitfaden für den PLM Markt, pp. 76–79. Weka GmbH, Darmstadt (2015b)
- Eigner, M., Gilz, T., Zafirov, R.: Neue Methoden, Prozesse und IT Lösungen für die virtuelle disziplinübergreifende Produktentwicklung. In: Berns, K., Schindler, C., Dreßler, K., Jörg, B., Kalmar, R., Zolynski, G. (Hrsg.) Proceedings of the 2nd commercial vehicle technology symposium (CVT 2012) (2012)
- Eigner, M., Roubanov, D., Zafirov, R.: Modellbasierte Virtuelle Produktentwicklung. Springer, Berlin (2014)
- Eisner, H.: Essentials of Project and Systems Engineering Management, 3rd edn. Wiley, New York (2008)

- Estefan, J.A.: Survey of Model-Based Systems Engineering (MBSE) methodologies. Technical Report. California Institute of Technology (25 May 2007)
- Forsber, K., Mooz, H.: The relationship of system engineering to the project cycle. Center for systems managements. Proceedings of the first annual meeting of the National Council for Systems Engineering and the 12th annual meeting of the American Society for Engineering Management, Chattanooga (20–23 October 1991)
- Freisleben, D., Schabacker, M.: Wissensbasierte Projektnavigation in der Produktentwicklung. In: Proceedings of the 20th CAD-FEM Users' Meeting 2002, International Conference on FEM-Technology, BD. 2, S. 1–10 (2002)
- Friedenthal, S.: Object oriented systems engineering. Process integration for 2000 and beyond: systems engineering and software symposium, 3rd edn. Lockheed Martin Corporation, New Orleans, LA (2014)
- Gajski, D.D.: Construction of a large scale multiprocessor. Cedar Project, Laboratory for advanced Supercomputers, Dept. of Computer Sciences, University of Illinois at Urbana-Champaign (Report/Department of Computer Sciences, No. UIUCDCS-R-83-1123) (1983)
- Gausemeier, J., Michels, J.S., Orlik, L., Redenius, A.: Modellierung und Planung von Produktentstehungsprozessen. In: VDI-Berichte Nr. 1819, S. 245–256. VDI-Verlag, Düsseldorf (2004)
- Gericke, K., Blessing, L.: Comparisons of design methodologies and process models across disciplines: a literature review. In: Culley, S.J., et al. (eds.) Design Processes, pp. 393–404. Design Society, Glasgow (2011)
- Gericke, K., Meißner, M., Paetzold, K.: Understanding the context of product development. Proceedings of the 19th international conference on engineering design 2013, 19–22. August 2013, Seoul, Korea (2013)
- Gericke, K., Moser, H.: Adapting a design approach: a case study in a small space company. In: Heisig, P., Clarkson, P.J., (eds.) Proceedings of 2nd international workshop on Modeling and Management of Engineering Processes MMEP, Cambridge UK (2012)
- Haberfellner, R., de Weck, O.L., Fricke, E., Vössner, S.: Systems Engineering: Grundlagen und Anwendung. Orell Füssli Verlage; 13. Auflage (2015)
- Hales, C., Gooch, S.: Managing engineering design, 2nd edn. Springer, London (2004)
- Hammer, M.: Seven insights about processes. In: Proceedings of the conference on strategic power process ensuring survival creating competitive advantage. Boston, USA (2001)
- Hitchins, D.K.: Systems Engineering: A 21st Century Systems Methodology. Wiley (2007)
- Horvath, I., Gerritsen, B.: Cyber-physical systems: concepts, technologies and implementation principles. In: Proceedings of TMCE 2012, Karlsruhe (2012)
- INCOSE: Technical operations. Systems engineering vision 2020, version 2.03. Seattle, WA: International Council on Systems Engineering, Seattle, WA, INCOSE-TP-2004-004-02 (2007)
- INCOSE: Systems engineering handbook, version 3.2; INCOSE-TP-2003-002-03.2 (2010)
- Kline, S.J.: Innovation is not a linear process. In: Research management, Vol. 26, Nr. 2, S.36–45 1995
- Kossiakoff, A., Sweet, W.N., Seymour, S.J., Biemer, S.M.: Systems Engineering Principles and Practice, 2. Auflage. Wiley (2010)
- Lee, E.: Cyber physical systems: Design challenges. In: 11th IEEE Symposium on Object Oriented Real Time Distributed Computing (ISORC) (2008)
- Lindemann, U.: Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden, 2. Auflage. Springer, Berlin (2007)
- Meißner, M., Blessing, L.: Adapting a design process to a new set of standards – a case study from the railway industry. In: Marjanovic, D. (ed.) 8th international design conference – design 2004. Design Society, Glasgow (2004)
- Miller, G.A., Galanter, E., Pribram, K.H.: Strategien des Handelns – Pläne und Strukturen des Verhaltens. Ernst Klett, Stuttgart (1973)
- NASA Systems Engineering Handbook. NASA, SP-610S, Washington (1995)
- NDIA, Wagner, Brockwell, Daniels, Loesh, Gosnell: NDIA Paper: use of a model-based approach to minimize system development risk and time-to-field for new systems. Final Report of the Model Based Engineering (MBE) Subcommittee NDIA, 10 Feb 2011 (2011)

- Negele, H.: Systemtechnische Methodik zur ganzheitlichen Modellierung am Bei-spiel der integrierten Produktentwicklung. Dissertation, TU München (1998)
- Oliver, D.W., Kelliher, T.P., Keegan, J.G.: Engineering Complex Systems with Models and Objects. McGraw-Hill (1997). ISBN:0-07-048188-1
- Omicciuolo, M., Thiel, M., Förster, K.P., Paetzold, K., Foerstner, R.: General purpose modeling and domain specific simulation: a framework for space mechanisms design. IEEE SysCon 2015, Proceedings, Vancouver (2015)
- Paetzold, K., Reitmeier, J.: Approaches for process attendant property validation of products. 1st International Conference on Modeling and Management of Engineering Processes MMEP 2010, Cambridge (2010)
- Pahl, G., Beitz, W.: Konstruktionslehre – Grundlagen erfolgreicher Produktent-wicklung. Methoden und Anwendung. 7. Auflage. Springer, Berlin (2007)
- Panreck, K.: Rechnergestützte Modellbildung physikalisch-technischer Systeme, Fortschritt-Berichte, VDI-Verlag, Düsseldorf (2002)
- Parraguez, P.: A networked perspective on the engineering design process. PhD-Thesis, Technical University of Denmark (2015)
- Patzak, G.: Systemtechnik – Planung komplexer innovativer Systeme, Grundlagen, Methoden, Techniken. Springer, Berlin (1982)
- Ponn, J., Lindemann, U. (eds.): Konzeptentwicklung und Gestaltung technischer Produkte. Springer, Heidelberg (2011)
- Pugh, S.: Total design: Integrated Methods for Successful Product Engineering. Addison-Wesley, Wokingham, England (1991)
- Rechtlin, E., Maier, M.W.: The Art of Systems Architecting. 2nd edn, CRC Press (2000)
- Reitmeier, J.: Eigenschaftsorientierte Simulationsplanung - Ein Beitrag zur effizienten virtuellen Absicherung der Produktfunktionalität. Dissertation, UniBW München, Verlag Dr. Hut (2015). ISBN:978-3843921626
- Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. The Addison-Wesley Object Technology Series. Addison-Wesley (1999)
- Sangiovanni-Vincentelli, A., Yang, G., Shukla, S.K., Mathaikutty, S.A., Sztipanovits, J.: Metamodeling: an emerging representation paradigm for system-level design. University of California Berkeley, IEEE Design & Test of Computers (2009)
- Skalak, S.C., Kemser, H.-P., Ter-Minassian, N.: Defining a product development methodology with concurrent engineering for small manufacturing companies. J. Eng. Des. 8(4), 305–328 (1997)
- Sop Njindam, T.: A systemic approach to analyze failures of complex multidisciplinary systems on the basis of their weak emergent behaviour. Dissertation, UniBW München, Verlag Dr. Hut (2015)
- Stachowiak, H.: Allgemeine Modelltheorie. Springer, Wien (1973)
- Sztipanovits, J.: Composition of cyber-physical systems. In: 14th Annual IEEE Int'l. Conference and Workshop on the engineering of Computer-Based Systems (ECBS '07), pp. 3–6. IEEE Computer Society, Washington (2007)
- VDI 2206: Entwicklungsmethodik für mechatronische Systeme, Hrsg. Verein Deutscher Ingenieure, Aug. Juni (2004)
- VDI 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, Hrsg. Verein Deutscher Ingenieure, Düsseldorf, Aug. November (1986)
- VDI 2422: Entwicklungsmethodik für Geräte mit Steuerung durch Mikroelektronik, Hrsg. Verein Deutscher Ingenieure, Düsseldorf (1994)
- Walker, R., Thomas, D.: A model of design representation and synthesis. 22nd design automation conference, Las Vegas (1985)
- Weber, C.: CPM/PDD – An extended theoretical approach to modeling products and product development processes. In: Bley, H., Jansen, H., Krause, F.-L., Shpitalni, M. (eds.) Proceedings of the German-Israeli symposium on advances in methods and systems for development of products and processes. Fraunhofer, Stuttgart (2005)
- Weilkins, T.: Systems Engineering mit SysML/UML: Modellierung, Analyse, Design, 2nd edn. dpunkt Verlag (2008)

Multi-Disciplinary Engineering for Cyber-Physical
Production Systems

Data Models and Software Solutions for Handling
Complex Engineering Projects

Biffi, S.; Lüder, A.; Gerhard, D. (Eds.)

2017, XII, 472 p. 138 illus., 82 illus. in color., Hardcover

ISBN: 978-3-319-56344-2