Chapter 3

Integrated Modelling of Information to Support Product Engineering Processes

A. Albers, A. Braun and T. Pinner

3.1 Introduction and Motivation

Product engineering processes are subject to increasing complexity. They comprise activities of product development, production and after sales such as service or decommission. Complexity arises from the large number of information elements and of their many interrelations (structural complexity - Maurer, 2007) in the context of product engineering. Information elements can be *e.g.* objectives with individual target values, activity description and their duration, decision criteria, resource capacities, *etc*. As a further challenge, uncertainty and dynamic behaviour of engineering processes lead to dynamic complexity (Diepold *et al*., 2010). In our research, we aim at handling the complexity of engineering processes through modelling information using the Integrated Product Engineering Model (iPeM - Albers and Braun, 2011). The iPeM provides a structure in which relevant information aspects can be clustered and interrelated. In this paper we present areas of potential support that can be realized with the iPeM modelling approach and present a prototypic implementation. We exemplify this concept by modelling selected aspects of a student project and use this test to evaluate our concept and to validate the software implementation. The paper is organised as follows: Section 3.1 outlines and motivates the research. In Section 3.2 we review the related state of the art and similar research approaches. From this, we substantiate why the iPeM is a suitable modelling framework for our research. Section 3.3 introduces the concept of our approach which is implemented as presented in Section 3.4. In Section 3.5 we describe an exemplary application which is critically discussed in order to evaluate our concept and to reflect upon the current software implementation. Section 3.6 concludes with a summary and an outlook on further work.

3.2 State of the Art

Table 3.1 gives an overview of selected approaches to modelling of product engineering processes. Exemplary aspects are compared to each other in this table which is in parts taken from Browning *et al.* (2006). The approaches can be classified by their respective focus on either design or project management or by their basis on either stages or activities (Wynn, 2007). Furthermore, the modelled elements and their typical variables and attributes can be distinguished. The different approaches contain diverse information contents depending on their modelling purpose.

Table 3.1. Overview of approaches to modelling of product engineering (PE) processes

Framework	Example References	Focus	Basis	Elements/ Contents	Variables/ Attributes
Activity Nets, PERT	(Elmaghraby, 1995)	Management	Activities	Tasks and their sequence	Activity duration elasticity
BPM	(Arkin, 2002)	Management	Activities	Activities objects	Myriad potential attributes
DSM	(Steward, 1981), (Eppinger, 2001)	Design or management	Activities	Activities and their relations	Dependency sequence
Integrated PD	(Andreasen and Hein, 1987), (Ehrlenspiel, 2007)	Design	Phases	Subsystems of PDP	Myriad potential attributes
IDEF, SADT	(NIST, 1993) (Ross, 1977)	Management	Activities	Function input, output	Control mechanisms
iPeM	(Albers and Braun, 2011)	Design and management	Activities and phases	Subsystems of PEP	Myriad potential attributes
Pahl/Beitz	(Pahl *et al*., 2007)	Design	Phases	Guidelines checklists	Product specification
Stage-Gate-Models	(Cooper, 2001)	Management	Phases	Stages, milestones	Stage duration decision critical
VDI 2206	(VDI 2206, 2004)	Design	Phases	Specification integration	Specification and validation criteria
VDI 2221	(VDI 2221, 1993)	Design	Phases	Stages, results for each state	Myriad potential attributes
ZOPH-Model	(Negele *et al*., 1997)	Design	Activities	Subsystems of PE processes	Myriad potential attributes

Most of these approaches are intended to serve distinct purposes, *e.g.* to establish transparency about activity relations or task sequences. Only ZOPH and iPeM have a holistic and systemic perspective on the system of product engineering. We apply the iPeM approach as a framework for our research since it aims at a holistic support of both designers and managers and considers the socio-technical nature of product engineering processes. The overall aim is to assist human beings in the centre of product engineering in terms of orientation, navigation, documentation, process- and knowledge work – with the help of transparent and integrated representation of information (Albers and Braun, 2011). Albers and Braun (2012) showed, that the iPeM allows modelling engineering processes at any necessary level of detail. It is based on system theory and can thus be regarded in a structural, hierarchical, and/or functional way (Ropohl, 1975). Hierarchic consideration allows clustering elements of related content and permits *e.g.* inheritance. Functional consideration helps representing the interconnectedness of the elements of the system through the exchange of deliverables (Albers and Braun, 2011). Changes on one single element exert influence on its interconnected elements and are propagated in the whole system of product engineering. For instance a change of “motor performance” may lead to changes on the “drive chain” (technical elements), but also to changes on organisational elements such as time schedules.

The iPeM meta model contains several subsystems and describes their interrelations. As also described by Ropohl (1975), a System of Objectives is transferred into a System of Objects by an Operation System. In the iPeM, the latter is further decomposed into a System of Resources and the activities matrix (Table 3.2). In this matrix, each activity of product engineering corresponds with a 7-step problem solving process (German acronym SPALTEN - Albers *et al.*, 2005). This forms a 10 x 7 matrix providing a structure for the assignment of information. The elements of the systems of objectives and objects as well as the system of resources’ elements may be interrelated with the activities in order to describe or prescribe functional dependencies; methods, but also knowledge and experience can be ascribed to the respective matrix field (Albers and Braun, 2011).

Table 3.2. Activities of the iPeM framework

Activities of Product Engineering	Activities of Problem Solving
Project planning and controlling	Situation analysis
Profile detection	Problem containment
Idea detection	Detection of alternative solutions
Modelling of principle solution and embodiment	Selection of solutions
validation	Analysis of consequences
Production system engineering	Deciding and implementing
Production	Recapitulation and learning
Market launch	
(Analysis of) utilisation	
(Analysis of) decommission	

In practice, several levels of product engineering processes can be distinguished *e.g.* planning or application. In the iPeM framework, activities can be arranged along a time bar in order to represent coherent phases or stages in a so-called phase model.

Here, a *reference model* depicts common invariant elements and their temporal dependencies describing past, similar engineering processes. It may represent best practice patterns that can be used to plan new projects. Such a plan results in an *implementation model*. An *application model* is the recording of a specific product engineering process showing the course of the real process. A set actual comparison can be used to readjust running projects or to learn from past processes in retrospect. The consideration of these model levels and the well-structured activities matrix separate the iPeM approach from other representations of the system of product engineering. We argue that this allows a wider range of support than a mere representation of activity or ZOPH-relations *e.g.* in a Multiple-Domain-Matrix (MDM) as presented by Hellenbrand and Lindemann (2011). Yet, it is still generic and could be applied more flexibly than approaches that focus on particular situations as for instance the pattern-based process navigator that has been developed by the research cooperation FORFLOW (Meerkamm *et al.*, 2009).

Albers and Braun (2012) showed in a test where a real project had been modelled descriptively, that the meta model of the iPeM is comprehensive enough to comprise any relevant information aspect in order to model engineering processes. However, the test also revealed limitations of the current (theoretical) state:

> *"The large amount of information leads to huge models fast, which requires additional means/possibilities for handling these representations by effective and efficient tools. A thorough investigation on the effort-value ratio has to be done before proceeding with any software implementation. For both modeling itself and for working with the models, usability needs to be enhanced."*
>
> (Albers and Braun, 2012)

In the next section we present a concept to enhance the iPeM's usability in practice with the aim of supporting product engineering.

3.3 Concept of the Integrated Modelling Approach

Our approach comprises the three model levels: reference, implementation and application level. Each of these levels may contain similar elements but represents different states of realisation. Where the application level represents the current AS-IS status, the implementation level contains the project-specific planning. The reference level contains planning elements that are project-unspecific and applicable for many different projects. In our consideration these levels are highly interconnected; every element may be included within one, two or three levels at the same time – which enhances the current understanding.

Every level contains five element classes according to the iPeM meta model: Objectives, Activities of Product Engineering, Activities of Problem Solving, Resources and Objects. All elements can be described in more detail by attributes such as durations in case of the activities. Elements can be combined with each other in order to describe particular dependencies within product engineering processes. In this paper, we focus on the prominent combination of the three elements Objective, Activity, Object, composing a so-called OAO-Triple.

According to the iPeM ontology, such triples describe how activities result from certain objectives and lead to related objects. In further considerations, also resources involved in this part of a process can be related. Not only objectives may be transformed into objects; also objects may lead to new objectives through activities such as analysis or validation. For instance, a strength calculation result of a shaft might lead to the awareness that a diameter or a steel grade needs to be changed.

3.3.1 Areas of Product Engineering Process Support

The following areas are adressed with our concept of integrated modelling.

Transparency of Dependencies is a concept to interpret and filter the holistic model in order to determine and to display relevant organisational and technical interdependencies for particular inquiries. This concept can be the means to cope with structural complexity. Due to multiple dependencies (often even across hierarchy levels), the effects of changes of elements cannot be foreseen directly. Calculating dependencies based on a holistic model can help to regain an overview for various purposes and omit mistakes due to oversight. A project manager might for instance be interested in interrelations of scheduled activities and resources. A designer might need transparency about the relation of technical product elements to elements of the system of objectives and so on.

Adaptive Project Control is a concept to support iterative planning and readjustment of processes by adapting to analysis results of the respective AS-IS status of projects or by adapting to changing boundary conditions. Thus, Adaptive Project Control can be a help to meet the challenge of dynamic complexity. Based on reference information that is assumed to be valid for a particular kind of project, specific implementation information can be established as a plan of the project. Monitoring application information (the project's AS-IS status), the planning state of the implementation level can be concretised or adjusted constantly. In this concept, there is a cycle of application and implementation that combines gradual planning with incremental realisation. Based on experience from transforming implementation into application level, successful planning information can again be stored as reference elements for other projects (Figure 3.1).

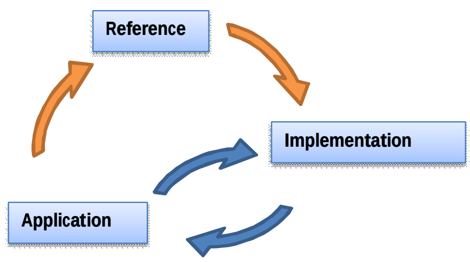

Figure 3.1. Circles of implementation and application within the three model levels

Best Practice Application is a concept of knowledge extraction and reuse. One example for a best practice pattern is the storage and provision of information about successful implementation to application transformations as described above. The information can be related to other elements in a distinct context. Our concept is to extract this knowledge from its carrier and to relate the information to generic iPeM elements (*e.g.* in form of OAO-Triples). With this, individual experiences can be modelled explicitly in a general framework. These representations may also contain individual boundary conditions of the respective situation; with this convenient retrieval and reuse of the knowledge in future projects becomes possible.

3.4 Software Implementation

We put our concept of supporting product engineering through a holistic modelling into practice with a software prototype. It is based on the CAM framework (Cambridge Advanced Modeller, see Wynn *et al.*, 2009). Information is put into the model manually at this stage of the prototype. We reflect on limitations of manual modelling later in the paper. Information is stored as an XML-file that comprises the model elements introduced in Section 3.3. The elements are visualised as follows in Figure 3.2.

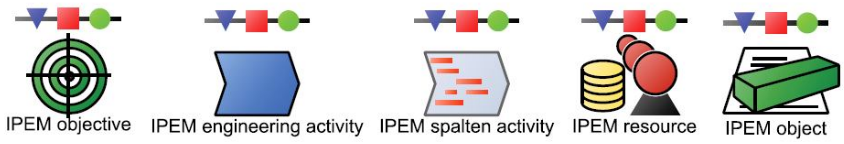

Figure 3.2. Visualisation of the elements in CAM

The connections on different model levels (reference, implementation, application) are visualised by different colours and connector shapes. Reference level connections are blue and shaped triangular; implementation level connections are red and indicated by square boxes; application level connections are green and feature a circle symbol. The following subsection introduces one particular view in which dependencies may be represented for different purposes.

3.4.1 DSM View

This view onto the holistic modelled data is based on a Design Structure Matrix (DSM, see Steward, 1981). Our DSM contains sub-systems, comprising the five model elements. The sub-systems are hierarchic, *i.e.* elements can be subordinate to other elements. With this, the level of detail of the representation may be adjusted to fit the respective purpose at hand. In the DSM, interconnections can be visualised across the sub-system boundaries. Figure 3.3 shows a screenshot of the DSM representation in

CAM where several model elements are connected in the three model levels as described in the paragraph above.

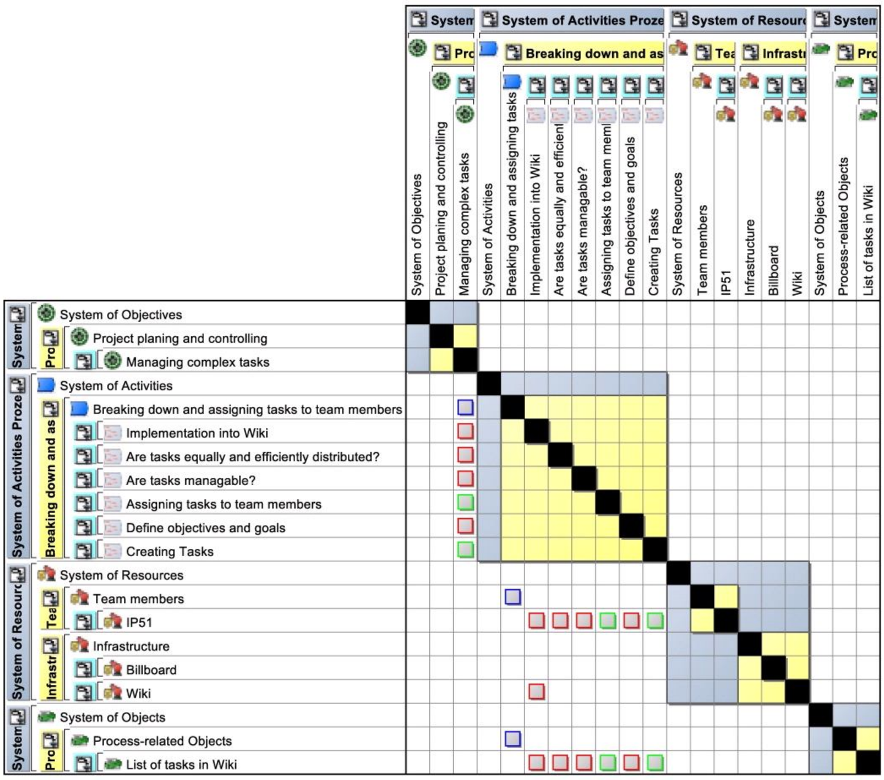

Figure 3.3. Screenshot of the DSM representation in CAM

3.4.2 Support through Transparent Dependencies

The DSM view is one tool aiming at transparency in the system of product engineering. In contrast to *e.g.* paper-based modelling, CAM offers several practical ways of further assistance. For instance, through a mouse-over user interaction, connected elements are directly highlighted which helps especially in navigation through large models.

Apart from that, several ideas for further assistance have been developed and implemented. Obvious but also hidden dependencies can be brought forward in specific perspectives called *explorer views* to achieve specific purposes. The objective explorer for instance (see Figure 3.4) uncovers dependencies between objectives, activities, resources and objects. Hereby, dependencies of technical aspects of the product can be represented in their interrelations; organisational aspects such as resource and activity planning are covered as well.

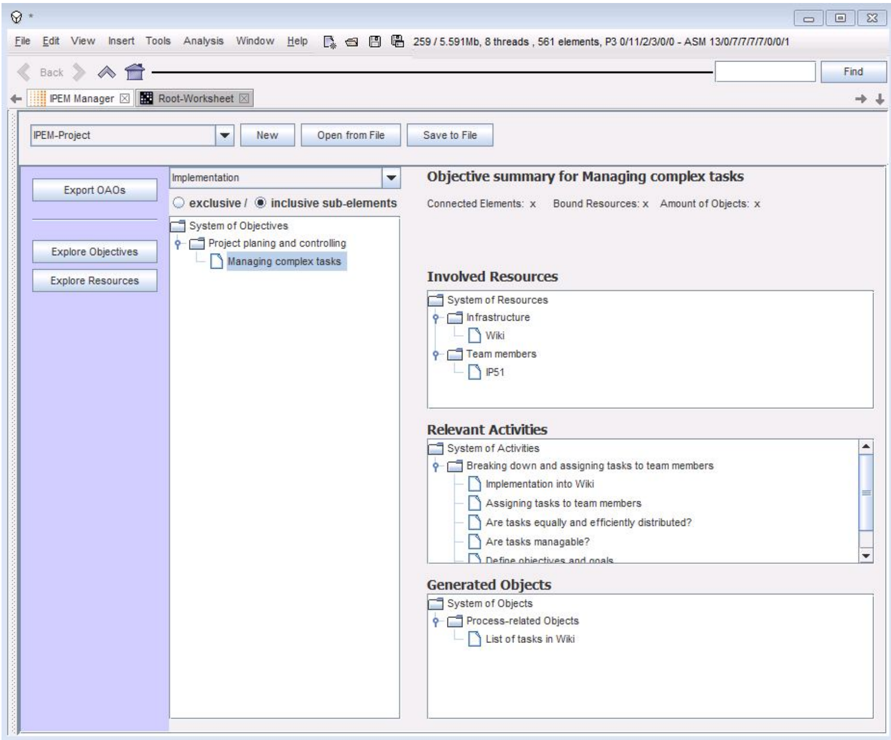

Figure 3.4. Explorer view with focus on objectives

Another representation that increases transparency - here with a focus on individual elements of the system of product engineering - is the *diagram view* where model elements are represented as boxes that are linked to each other via arrows. With the help of this, dependencies can be explored intuitively by selecting single elements of the diagram in order to optionally show their direct and/or indirect relations. A single click on any element focuses the view on this and shows all its connected elements. Double-clicking an element allows expanding or collapsing it in order to explore its hierarchic relations as well. In order to adjust the view to a given problem at hand, it is furthermore possible to select the element classes that shall be displayed in the diagram. Their hierarchy level may also be selected in order to further customise views.

Transparency established by integrated modelling also helps to develop projects in a managerial view. Through interrelating the three model levels of the iPeM (see Figure 3.1) it is possible to continuously validate a current process and to adapt it to changes; best practices can be stored and reused at all times. In a wider perspective, the core idea of this approach is to systematically reduce uncertainty that stems from the structural complexity and the dynamic nature of product engineering processes. For *Adaptive Project Control* purposes in the DSM view, the three model levels of the iPeM are indicated by colours and can be specified when adding new elements.

In our prototype, best practices can be exported by selecting existing combinations of objectives, activities and objects. With a particular user interface it is possible to make information available to other users. *Best practice OAO-Triples* can be selected from a list in order to export them as an XML-file. Every triple can be described through tags to facilitate the detection and reuse of suitable reference information in other projects. Users can select multiple reference elements and build individual reference patterns. As a result, the project's final reference level is geared towards the specific project and with regard to the combination of multiple reference elements. The idea of generally usable references but individualised support is realised in this way. Thus, the import and export of reference elements integrates seamlessly into the concept of adaptive project planning.

3.5 Exemplary Application in a Student Project

In this section we present a first application of our approach. The class 2011/2012 of the academic course "Integrated Product Development" has been chosen as a use case for the exemplary application. It is a four month product engineering project with a leading industrial partner and takes place in a realistic environment. It includes all stages of a (totally) new design - all the way from the definition of the market niche to the production of functional prototypes - as well as project management (time, budget, *etc.*). The project phase of one group of six students, where market demands have been detected and described, was modelled for our test. The project's initial task description in IP is very vague; hence uncertainty is particularly high. Therefore, it serves well for the purpose of an evaluation of our support approach. At the same time, the entire process is well observable as the supervisors have access to all intermediate files, sketches, documents, project plans, *etc*.

The project has been attended by a graduating student who is working on his thesis on product modelling. The model is comprehensive and includes over 800 elements with their attributes and connections to each other. For an application in industry, however, modelling by an external person can be a notable restraint, as described later in this chapter.

3.5.1 Evaluation of the Concept

The DSM view shown in Figure 3.3 contains excerpted elements of the four hierarchic subsystems according to the iPeM ontology. This provides a clear structure, in which the information elements can be modelled. The concept of interrelating reference, implementation and application level information is put into practice as follows in IP. For instance an exemplary objective - the need to manage complex tasks - arose during the project (see column 3 in Figure 3.3). A potential reference approach to deal with this objective is represented as a group of activities with the collective name *"Breaking down and assigning tasks to team members"* in our example. Reference information such as this can be provided by the project supervisors who assist the students in IP based on their own experience. In a first modelling step the (reference) activities lead to unspecified *"Process-related objects"* (see reference connections in column three/line five and in the fourth column of the DSM).

This reference model could be specified more precisely when knowledge about the project increased; *i.e.* a planning at a deeper level of detail was performed (modelled on implementation level). Sub-elements of the existing activity were defined and assigned to a more specific object (columns three and columns six to eleven). Nonetheless, these activities are still related to the same objective "Managing complex tasks". Consequently, this objective now belongs both to the reference and the implementation layer. In the further course of the project, performed activities were recorded in the application layer in which the real happenings and resulting objects are captured. By doing so, one could also store experiences as short text descriptions or hyperlinks to a product data management system, *etc*. The example shows this third type of connection between the objective "Managing complex tasks" and the activities "Create tasks" and "Assigning tasks to team members". Furthermore, the actual executor of the activities is visible (connection to resources). Successful combinations of real-life-proven procedures can be shared directly as reference which changes their signification from recorded data to guidelines for other projects.

This example shows that the transparent and adjustable views on the modelled information can be used successfully in a practical application. With this, the concepts of Adaptive Project Control and - in parts - also knowledge reuse could be exercised and evaluated. Feedback from the students whose project had been modelled, and a critical consideration of the insight from the field test by the modeller and the authors of this paper indicate that the concepts for support of product engineering processes presented in Section 3.3 work well.

However, there is also a critical reflection on the current software implementation: the software prototype in CAM is not meant to serve as a marketable computer programme. It was designed to support the concepts described above with the aim to allow a first evaluation. One big restraint is that a multi-user assistance has not been realised yet. Another problem is the effortive acquisition of information. Apart from the required time, also corruption of information due to the modelling by a third person hinders the benefit of the approach today. Further work should address ways to get large parts of model "on the fly" during running projects - *e.g.* with the help of tracking tools.

3.6 Conclusions and Outlook

In this final section we provide a summary of the approach and the findings of the case study. We close with an outlook on further research directions and work to be done considering software implementations of the iPeM.

3.6.1 Summary

In this paper, an approach for a methodological support of product engineering processes has been presented. After reflecting on the state of the art considering background and literature about available modelling approaches, own concepts for project support based on the iPeM have been presented. We illustrated a prototypic software implementation in CAM with the help of which these concepts have been put into practice. The application of this tool in a student project served as an exemplary use case and proved the approach's potential for a support of product engineering processes. Even with the limited scope of operation, the prototypic implementation helped well to model aspects of the student project in terms of the iPeM. Compared to pen and paper based approaches for instance, the software tool facilitates model creation and handling - especially for large models. However, there are several open questions (*e.g.* semi-automated modelling) and also weaknesses to be worked on in future efforts.

3.6.2 Outlook

The methodology for support presented in this paper is limited to Transparency, Adaptive Project Control and Best Practice Application for a first evaluation. Apart from that, a broad range of further support could be realised. Saak (2007), for instance, described a concept for a computer-aided tool for the efficient employment of the problem solving methodology "SPALTEN". It provides methodological support for each of the SPALTEN steps and would therefore benefit the iPeM application in practice. In a next step, consequently, this concept can be integrated in our software implementation.

DSM representations are based on graph theory. Here, comprehensive analysis methods can be applied to the selection of elements of interest. Prominent examples would be communication path analysis or critical path analysis for a schedule of activities. Dependencies between objectives or objects (*e.g.* contradictoriness of objectives or calculation results) might be analysed as well (see Browning, 2001 for DSM analysis methods).

The current software prototype also needs further effort in order to increase its usability for general use since it was only developed for our range of applications. Especially restrictions of the export or import functions have to be mentioned here. The intention to apply the tool *e.g.* in industry without experienced modellers or the application by students in a wider research study leads to open questions considering handling of information. We could show that our approach offers several beneficial functionalities; however the support can only be as good as the information it is based on. Therefore and most of all in order to reduce effort, ways to acquire data and to put it into the model efficiently should be developed.

3.7 References

Albers A, Braun A (2011) A generalized framework to compass and to support complex product engineering processes. International Journal of Product Development, 15(1/2/3): 6-25

Albers A, Braun A (2012) Towards handling complexity - Testing the iPeM process modeling approach. In: Proceedings of the 10th International Symposium on Tools and Methods of Competitive Engineering (TMCE 2012), Karlsruhe, Germany

Albers A, Burkardt N, Meboldt M (2005) SPALTEN problem solving methodology in the product development. In: Proceedings of the 15th International Conference on Engineering Design (ICED'05), Melbourne, Australia

Andreasen MM, Hein L (1987) Integrated product development. IFS (Publications) Ltd and Springer-Verlag, London, UK

Arkin A (2002) Business process modeling language (BPMI). Available at: http://www.omg.org/bpmn/Documents/BPML-2003.pdf (Accessed on 1st November 2012)

Browning TR (2001) Applying the design structure matrix to system decomposition and integration problems: A review and new directions. IEEE Transactions on Engineering Management, 48(3): 292-306

Browning TR, Fricke E, Negele HE (2006) Key concepts in modeling product development processes. Systems Engineering, 9(2): 104-128

Cooper RG (2001) Winning at new products: Accelerating the process from idea to launch. Perseus Publishing, Reading, MA, US

Diepold K, Biedermann W, Eben K, Kortler S, Lohmann B *et al.* (2010) Combining structural complexity management and hybrid dynamical system modelling. In: Proceedings of the 11th International Design Conference (DESIGN 2010), Dubrovnik, Croatia

Ehrlenspiel K (2007) Integrierte Produktentwicklung. Hanser, München, Germany

Elmaghraby SE (1995) Activity nets: A guided tour through some recent developments, European Journal of Operational Research, 82(3): 383-408

Eppinger SD (2001) Innovation at the speed of information. Harvard Business Review, 79(1): 149-158

Hellenbrand D, Lindemann U (2011) A framework for integrated process modeling and planning of mechatronic products. In: Proceedings of the 18th International Conference on Engineering Design (ICED 2011), Copenhagen, Denmark

Maurer M (2007) Structural awareness in complex product design. Verl. Dr. Hut, München, Germany

Meerkamm H, Henrich A, Jablonski S, Krcmar H, Lindemann U *et al.* (2009) Flexible prozessunterstützung in der produktentwicklung: Prozesse - daten - navigation. Shaker Verlag, Aachen, Germany

Negele HE, Fricke E, Igenbergs E (1997) ZOPH - A systemic approach to the modeling of product development systems. In: Proceedings of the 7th Annual International Symposium of INCOSE, LA, US

NIST (1993) Integration Definition for Function Modeling. IDEF0. National Institute of Standards Technology, National Technical Information Service, US Department of Commerce, Springfield, VA, US

Pahl G, Beitz W, Feldhusen J, Grote K-H (2007) Engineering design - A systematic approach, 3rd edn. Springer, Berlin, Germany

Ropohl GH (1975) Systemtechnik - grundlagen und anwendung. Carl Hanser Verlag, München, Germany

Ross DT (1977) Structured analysis (SA): A language for communicating ideas. IEEE Transactions on Software Engineering SE, 3(1): 16-34

Saak M (2007) Entwicklung eines konzeptes und eines prototypen für ein rechnergestütztes werkzeug zum effizienten einsatz der problemlösungsmethodik SPALTEN. IPEK Dissertation, Universität Karlsruhe Institut für Produktentwicklung, Karlsruhe, Germany

Steward D (1981) The design structure matrix: A method for managing the design of complex systems. IEEE Transactions on Engineering Management, EM-28(3): 71-74

VDI 2206 (2004) Entwicklungsmethodik für mechatronische systeme. Beuth Verlag, Berlin, Germany

VDI 2221 (1993) Methodik zum entwickeln und konstruieren technischer systeme und produkte. Beuth Verlag, Berlin, Germany

Wynn DC (2007) Model-based approaches to support process improvement in complex product development. PhD Thesis, Cambridge Engineering Design Centre, University of Cambridge, Cambridge, UK

Wynn DC, Nair SMT, Clarkson PJ (2009) The P3 platform: An approach and software system for developing diagrammatic model-based methods in design research. In: Proceedings of the 17th International Conference on Engineering Design (ICED'09), Stanford, CA, US