INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

A PRACTICAL APPROACH TO STRUCTURE THE PRODUCT DEVELOPMENT PROCESS USING NETWORK THEORY

A. Chahin, J. Hoffmeister, K. Paetzold, N. Noori and X. Vilasis Cardona

Keywords: networks, Petri nets, FORFLOW-process-model, complexity management, product development process

1. Introduction

The integration of various functions into one product within a short-time product development process (PDP) is impeding to the coordination processes and the optimization of PDP itself. There is a need for generic framework that is adaptive to the companies' situation or the development tasks. Most common theoretical approaches for illustrating PDP's are flow diagrams or directed graphs. These are oversimplified and not detailed enough to provide complete descriptions of data flow, information exchange and decisions made in order to enhance existing business processes. The usage of those methods requires further adaptations depending on the individual corporate structure. There are other PDP models used by companies that can be too complex to be retraceable. Another PDP representation is the FORFLOW-Process-Model (FFPM) is used as well. FFPM is a result of cooperative work between scientists and 20 companies from the industry that created a compromise between practical and theoretical perspectives. The first mapping of the FFPM as a network (NW) proved that interfaces should be defined during early stages of PDP design and NW mapping in one layer is inadvisable. The facilitate managing and manipulating the PDP, NW should be modeled in a multilayer approach [Parraguez 2015]. Neither of the PDP models, V-Model or FFPM, which are currently used give any feedback about tasks dependencies nor show impact of artifact, like CAD-Data Requirements, or impact of the company structure on the PDP. Moreover, it is not possible to check impact and timing of each artifact before, during or at the end of the PDP. Consequently, there is a need for methods to model the impact of artifacts and to offer an alternative interpretation, which influences the PDP, in order to set the constructions. In addition, the lack of feedback impact the developers' ability to stay updated about the progression during the PDP. The existing methods are not detailed enough for such usage as they are generally static and do not provide a dynamic representation.

In this paper, we utilize FFPM as a representative PDP, framework for interfaces' definition and illustrate how a PDP network (NW) can be separated into multiple layers and how to use Petri-Nets (PN) to describe each of the single actions of the PDP. A practical application of a PDP model based on Network Theory (NWT) that can describe the flow of information in the process. Such representation would support the decision making process and enables developers to define interfaces at a very early stage in the design process. The proposed model contributes advance the enhancement of PDP as it provides clarity of the requirements of the design process itself and how dependencies can be changed and have more details to support further progress requirement engineering.

2. Research question

There is a persisting need for a PDP model that can meet the practical demands of the industry. Those models need to fulfill certain conditions in order to support the decision process. Such models must be:

- Generic and dynamic
- Represent the complete data- and info flow

Hence the question is, how to build a holistic, dynamic and NWT based PDP model to fulfil the practical industry requirements mentioned above?

3. State of the art

Before describing the desired framework, it is important to know how the NW's are being used for modeling PDP's so far. In the next sections, the process will described and then followed by steps of NW construction.

3.1 Product-Development-Process

In our work, the FFPM serves as a representative development process. It is a generic PDP based on common steps and stages referring to the VDI 2221 and the V-Model. It includes all known general phases of the PDP - from the initial idea to the start of the production [Rieg et al. 2009]. The FFPM is divided into six steps on the abstract level (cf. Figure 1 – Level 1), which become more detailed in progress. Using sub processes makes the model more explicit and leads to 90 steps divided in six phases (cf. Figure 1- Level 2 and level 3):

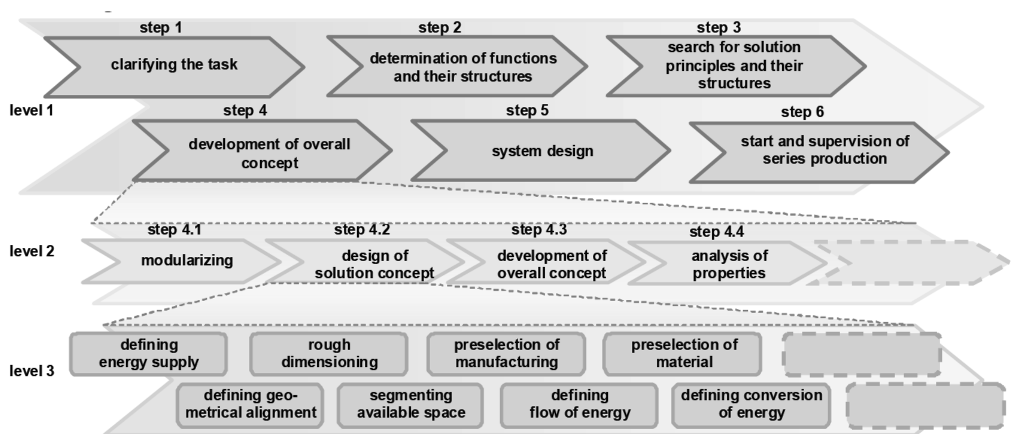

Figure 1. The FORFLOW-Process-Model [Krehmer 2012]

The typical procedure used in product development, like the model of the VDI 2221, is represented in the first - generic - level. According to the VDI 2221, the second level allows to model specific steps following a strict timeline. The second level is already adaptable its own boundary conditions such as the market, branch or competition. In the third level, tasks are more specific and the necessary activities are now being defined. In this level it is important to maintain a logical timing of the sequence.

3.2 Using networks for modeling the PDP

The focus of modeling NWs relies on describing the flow of information and data. Many approaches dealing with this challenge are mentioned in the literature. One of these frameworks is done by Weber. It describes the product model by using dependency of properties and characteristics to build the CPM/ PDD-Approach. This is used by [Weber 2005] to fill the so-called Multi-Domain-Matrix during the PDP describing the dependency on the properties and characteristics. Using this matrix allows to see the progression of the PDP as well as allows for process classification and allows mathematical analysis.
To allow maximum benefit from Weber method, a NW segmentation is required. It is reasonable to divide the process into three layers: Artifacts, corporate structure and process – cf. Figure 2.

This separation represents the complete NW and shows every connection between the layers. Having a multi-layer representation helps preserving the overall information of the system. The multilayer representation provided a detailed visualization that enables performing analysis inside a certain level and between two different layers.
Parraguez shows the dependency between level two and three by modeling a NW [Parraguez 2015]. By using this NW, it becomes possible to retrace, which tasks are done by which unit or team. The second level is very detailed in a way that allows analyzing the work of every single employee and its team. This makes it possible to review every team's accomplishments and to compare them to other team's achievements based on the quality and speed of the performed tasks.

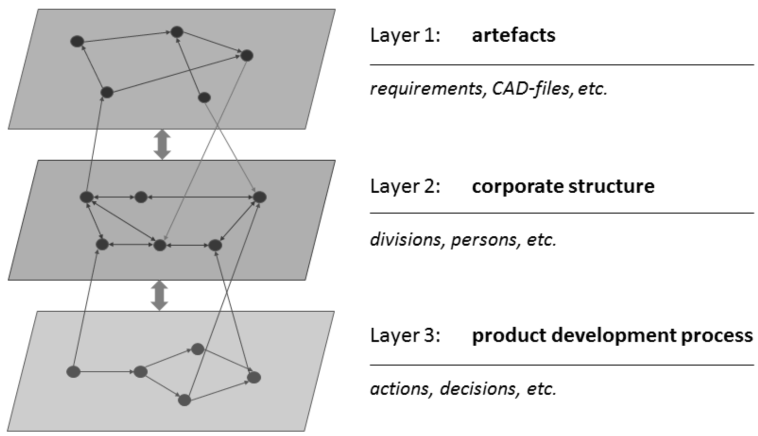

Figure 2. Three levels of network representing the product development process

NWs are used in the case of crisis management, too. The methods are similar: A NW is created and analyzed representing the dependencies between different organizations and missions. This helps to setup teams in an efficient way and to solve a crisis within a shorter time period [Noori 2015].
This usage of NWs is similar to Parraguezs. It is operating between the second and third layer.
The signposting framework of Wynn, Eckert and Clarkson [Wynn et al. 2006] describes the relation between tasks of the third layer and parameters of the first one. These parameters refer to values of the product, which is a subset of the first layer. In order to have a complete description on the first layer, it is necessary to consider all artifacts needed and not only the product data.
The so called traceability - or more common the dependency - between two nodes is used by Figge to model the connections within the first layer [Figge 2014]. By setting up a NW, he shows the component dependencies of the product. Analyzing this NW clarifies the interfaces between the components and makes it possible to estimate the responsibility of each department. The algorithm used here are comparable to the ones used by Parraguez [2015].

3.3 Petri-Nets

Petri nets - as developed by Carl Adam Petri [1962] – is an even based discrete system method that can be utilized for describing distributed systems. A Petri Nets (PN) is a bipartite graph, separated into places and transitions. A place represents a condition of the system, while a transition stands for an action/event. Since a PN is a bipartite graph, a place can only be connected to transitions or vice versa.
A place may contain a number of tokens. Any possible distribution of tokens within the net is called marking. The state of the system is described by these markings. An arc leading to a transition defines a condition for it to be fired (i.e. an action to be performed). A transition can only fire, if the place leading to it has a sufficient amount of tokens assigned, defined by the weight of the arc.
The PN shown in Figure 3 consists of three places (circles) and one transition (square). The transition can only fire, if p1 and p2 each contain one token. It will then create a token in p3. However, the condition is not fulfilled, since no token is assigned to p1. Thus the system is deadlocked.

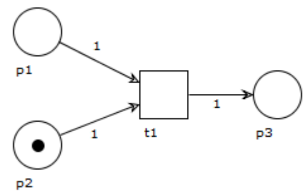

Figure 3. Example of a simple Petri Net created with WoPeD

This way, a PN is able to describe a dynamic system by its possible markings. There are several extensions of PNs such as timed PNs, which were developed in order to describe certain systems more accurately [Ramchandani 1973].

3.4 Current gap

In conclusion, we recognized that the gap becomes clear where missed connections between the first and third layer of the NW model. Up to now there is no connection between all artifacts and the process. The dependency of the single tasks on the artifacts is neither given nor analyzed. A model showing all the artifacts' influence on the PDP is not yet built. For instance, there is no way to analyze the influence of the requirements on the PDP.

4. Suggested approach

The segmentation of the PDP into three levels must be kept in this approach. The nodes have the same characteristics within each layer. This means, for example, that the requirements or the CAD-data on the first layer are artifacts and can possibly contain sub-networks (SNW). The nodes and edges on the first layer might however be weighted in another way than those on the second layer, i.e. the meaning of the weights differ from each other. This opens new possibilities of analysis. The layers, however, have to be treated as separate NWs and the edges connecting each layer need to be weighted in yet another way. The addition of SNWs gives the possibility to have more details and to allow an optional level of classification. This method of detailing and the hierarchical classification - especially used in layer two - represents the whole PDP within one NW and minimizes the danger of losing information by reducing the complexity of the complete NW. Using the FFPM and the methods of the graph theory enables the modeling of undirected NWs that show the tasks sequence of the PDP. These are connected to each other and end up in intersections. Modeling the whole PDP in one NW appears not to be very useful. With its high number of nodes it becomes too complex [Chahin and Paetzold 2015]. Additionally, it was not possible to analyze the NW mathematically, because the contained nodes have different characteristics. The major transformation is required in the third layer of the NW. PN (cf. Figure 4) are applied for building up the tasks.

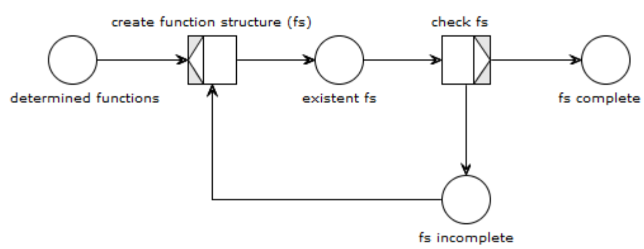

Figure 4. Part of a Petri-Nets, modelled with WoPeD

This means that a task is represented by a single node inside the third layer of the complete NW. These nodes are linked to each other according to the sequence of tasks. One task summarizes special jobs of the PDP, which are combined with each other into a PN that represents the workflow and makes the NW in this special nodes dynamic. Using PN gives new opportunities of analytical methods and possible results.

5. Framework of the network

Before starting to build up the complete NW, it is important and necessary to extend the mathematical methods in order to have a wider field of analytical possibilities. After that, the third level of the NW is build up. The advantages of PN are explained afterwards. Thereby, the third level is extended. At the end, the new possibilities are exemplified.

5.1 Mathematical extension

NWs are composed of nodes linked to each other by edges. The nodes are typically labeled or have a unique identification. The edges could be directed or non-directed and are also extendible by weighting. A NW is generally transferable into a matrix and vice versa [Krischke and Röpcke 2015]. According to the standard, tools like NodeXL, Pajek, etc. already support this function. Each entry of the matrix defines a link between two nodes. The following Figure 5 shows two nodes, A and B, which are connected to C. The matrix shows how the translation looks like. The link is represented by an x, which could also be a number that clarifies a weight.

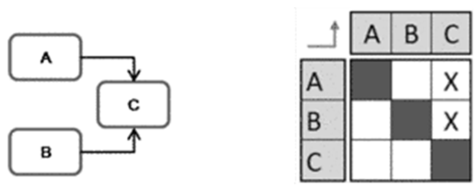

Figure 5. An example of a Network and the translation into a matrix

The method explained above is used to describe a NW as a graph or a matrix and is not yet sufficient. In order to close the gap, an extension of the mathematical methods is needed.
The most essential upgrading is done by changing the items of the matrix. As explained, the entry of each field represents a directed connection and is a scalar until now:

$$x_{i,j} \in \mathbb{Z};\ i, j \in \mathbb{N}. \tag{1}$$

Therefore, a field of the matrix could only carry one information. To enlarge the possibilities the following definition is introduced:

$$x_{n,i,j} \in \mathbb{Q};\ n, i, j \in \mathbb{N} \tag{2}$$

Instead of giving an edge only a single weight, it is now possible to add an optional amount of information to a single edge.
Almost the same is done to the diagonal. Every input of the diagonal is defined as:

$$x_{n,i,j} \in \mathbb{Q}\ for\ i = j \tag{3}$$

This means that a node carries not only its identification, but could also be extended with additional information. Based on the needed application, the nodes are extended by the scalar "time" that is needed to complete a task. It can be in minutes, hours or similar. This additional information is going to be helpful in order to find out, how long it takes for a task to be performed and which tasks are slowing down the PDP. Layer 1 can now carry the progress as a further information. The weighting of the second layer will define the size of each team.
It is of Corse possible to separate the n-tuple of the matrix in n-matrix. But having all the information in one matrix is clearer and reduces the amount of data - cf. Equation (4):

$$X = \begin{bmatrix} (x_1, x_2)_{1,1} & \cdots & (x_1, x_2)_{1,j} \\ \vdots & \ddots & \vdots \\ (x_1, x_2)_{i,1} & \cdots & (x_1, x_2)_{i,j} \end{bmatrix} \tag{4}$$

5.2 Structure of the complete network

As mentioned above, the NW is build-up of the three NWs - separated in three layers.

The first layer includes all the information about the artifacts. The main NW is detailed by several SNWs, which can be dynamic and static:

- Requirements
- CAD-Data (separated in components)
- Guidelines (VDI, DIN, ISO, etc.)
- Provisions and physical law
- Project plans

The second layer of the whole NW is about the internal structure of a company and is hierarchically composed of quasi-statically SNWs:

- Company, Supplier
- Divisions, Teams up to the single Employer

The Divisions are separated in the three main domains: Mechanical and electrical engineering and computer sciences.

The third and most dynamical NW represents the process. The nodes combine tasks that belong to interlines or to a certain milestone. The main NW is a directional graph, while the tasks are modeled into several PNs.

5.3 Using Petri-Nets in the Product-Development-Process

The PDP has the character of a workflow, as defined in [Van der Aalst 1998]. For that reason, PNs appear to be suited for the mapping of the PDP.

At this point however, it is not possible to integrate the PN into the superordinate network - cf. Figure 6. Therefore they will be treated as SNWs within layer 3 (PDP). These SNWs will be analyzed separately and allows to have several stages of the NW. The PN is indeed the dynamic representation of n-time static NWs [McMahon and Xianyi 1996]. The results can then be used as the weights of nods and edges within the layers 1 and 3 of the overall network.

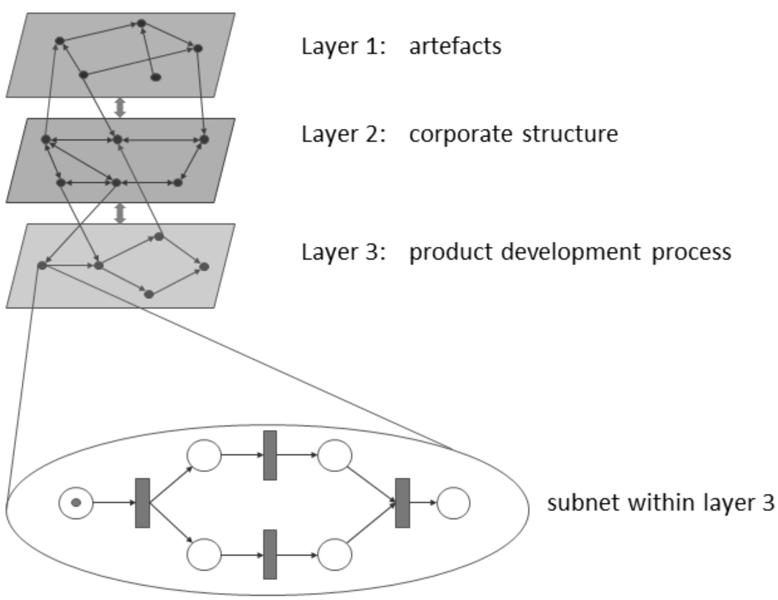

Figure 6. Petri-Nets integrated into the superordinate network (layer 3)

The marking of a PN, i.e. the state of one subtask of the PDP, depends on the progress of specific artifacts. Certain tasks can only be performed, if an artefact (such as the CAD-files of a product) is in a sufficiently advanced state. By analyzing the PNs, the importance of artifacts for each task can be identified to a certain degree. In return, the performance of a task can influence the progress of artifacts. This information will be used to weight the edges of the superordinate network.

That way, it is possible to determine the significance of each task within the PDP regarding layer 1.

As an example, the list of requirements might be used several times for one task of the PDP. The importance of the task (regarding this artifact) could be identified by looking at the influence the list of requirements has on its subtasks (represented by a PN). The artifact will be used a certain number of times and the completion of the task will likely come with changes to the document.
Furthermore PN enable the expansion of the network by time management. While this is important for capacity planning, it plays a subordinate role in the weighting of the network. The analysis of the time required for the performance of tasks can however be used to prioritize subtasks within concurrent processes of the network.

5.4 Advantageous of Petri-Nets

PNs are generally well-suited for the modeling of workflows, which are commonly used in the presentation of business procedures. This is due to their ability to represent concurrent processes and to enable the analysis regarding critical information needed for operative planning (e.g. deadlocks).
It is possible to describe preconditions for a task to be performed and to include if/or-queries in the model. Figure 7 shows an example of a concurrent workflow process. Task D can only be started after the completion of the tasks B and C.

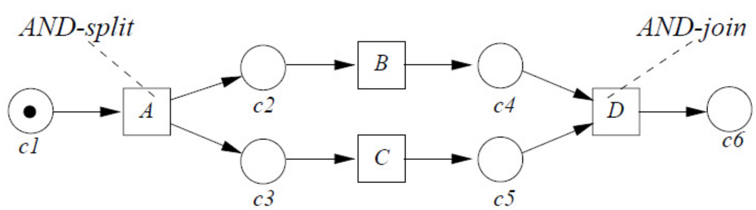

Figure 7. Example of a concurrent process modeled as a Petri-Net

Timed PN furthermore allow for a time management system within the workflow. This opens up possibilities to expand the model by capacity planning regarding layer 2 (corporate structure).

6. Conclusion of the approached network

In Figure 6, we demonstrated the concepts of separating the NW in three layers and adding PNs to the PDP to allow for new ways to determine NW properties, such as cluster coefficients, centrality values, robustness, etc. It is also possible to weight the edges connecting the layers one and three.
Applying the NW to a running PDP provides a feedback regarding the expenditure of time, which is important additional information. Furthermore, it is possible to determine the progress of an artifact and the influence it has on certain tasks. This way, priorities of different tasks can be identified.
In the case of developing variants or adoptions, the analysis of the former procedures will deliver estimation on how much time the overall process will take. Depending on the given information it is even possible to simulate the PDP and to verify its feasibility as a result.
This is followed by the extensions of mathematical methods as stable base and is widening the opportunities. Additionally, PNs are used to dynamically model the tasks within the overall NW.
To sum up, these aspects need to be considered in order to build a suitable model of the PDP that allows the usage of NWT. Thereby, it is now possible to see the influence of the artifacts on the PDP and to check how long it takes to achieve a certain progress within the PDP.
Moreover, critical tasks can be detected, which potentially slow down the process. The state of an artifact advances depending on previously performed tasks.

7. Outlook

In order to give a respectable statement, it is necessary to run this approach on a real PDP. Using discrete data allows to use mathematical methods. This way it is possible to prove the viability of n-tuple instead of a single scalar.
Recommendations to improve the process can be deduced based on the results of algorithms. In case of a running company, data of previous PDPs can be analyzed with the NW in order to detect tendencies

regarding their performance. This will provide the necessary parameters for running a simulation of a future PDP, which in return gives feedback about its estimated runtime.
Furthermore, there are many applicable algorithm declared in the theoretical graph- and network theory, which are given and cannot be run on the theoretical framework.
A further challenge is the addition of the methods mentioned in chapter 3 (state of art) into the suggested NW. This requires a more detailed mapping of the corporate structure including the hierarchical levels of the company. This would give a more realistic view of the complex system and allow the planning of capacities such as working hours. Expanding the model by the approach described in [Parraguez 2015] gives an idea of the optimal structure of a department. As a conclusion the overall NW would provide a complete view of the PDP containing all relevant information and data of the process. The application of the model to a real company is expected to significantly improve the efficiency of product development.

References

Chahin, A., Paetzold, K., "Analyse des Produktentwicklungsprozesses mit Methoden der komplexen Netzwerk-theorie", Proceedings of the 26th Dfx-Symposium, Krause, D., Paetzold, K., Wartzack, S. (eds.), TuTech Verlag, Herrsching, 2015.
Figge, A., "Effiziente Erfassung und Pflege von Traceability-Modellen zur Entwicklung technischer Systeme", Fraunhofer Verlag, Berlin, 2014.
Krehmer, H., "Vorgehensmodell zum Iterations- und Produktreifegradmanagement in der eigenschaftsbasierten Produktentwicklung", VDI-Verlag, Düsseldorf, 2012.
Krischke, A., Röpcke, H., "Graphen und Netzwerktheorie", Hanser Verlag, Munic, 2015.
Kunc, M., "System Dynamics and Innovation: A complex problem with multiple levels of analysis", Proceedings of the 30th International Conference of the System Dynamics Society, St. Gallen, 2012.
Mamorot, M., Marchlewitz, S., Winzer, P., "Changeability of Requirements - Useable Indicators for Product Development", Proceeding of IEEE International Conference on Mechatronics (ICM), Vicenza, 2013, pp. 350-355.
McMahon, C., Xianyi, M., "A network approach to parametric design integration", Springer-Verlag London, 1996, 14-32.
Parraguez, P., "A Networked Presprective on the Engineering Design Process", PhD Thesis, Techinical University of Denmark, 2015.
Parraguez, P., Maier, A. M., "Unfolding the design process architecture: A networked perspective on activities",Proceedings of the 15th ICED, Milan, 2015.
Petri, C. A., "Kommunikation mit Automaten", PhD Thesis, Faculty of mathematics and physics, Technical university of Darmstadt, 1962.
Ramchandani, C., "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets", Technical report, Massachusetts Institute of Technology, Cambridge, 1973.
Reitmeier, J., Chahin, A., Paetzold, K., "An approach to the property-based planning of simulations", Proceeding of 20th ICED, Milan, 2015.
Reitmeier, J., Luft, T., Paetzold, K., Wartzack, S., "Consideration of Uncertainties in the Product Development Process", Chapter 12 of Modelling and management of engineering process, Springer-Verlag, Berlin, Heidelberg, 2015, pp. 137-149.
Rieg, F., Merkamm, H., Henrich, A., Jablonski, S., Krcmar, H., Lindemann, U., "Flexible Prozessunterstützung in der Produktentwicklung: Prozesse -Daten –Navigation", Shaker Verlag, Aachen, 2009.
Van Der Aalst, W. M. P., "The application of petri nets to workflow management", Journal of Circuits, Systems and Computers, Vol. 8, 1998.
Weber, C., "CPM/ PDD - An extended theoretical approach to modeling products and product development processes", Proceedingsof the 2nd German-Israeli Symposium, Stuttgart, 2005.
Wynn, D., Eckert, C., Clarkson, P., "Applied signpostin: A modeling framework to support design process improvement", Proceedings of IDETC/CIE, Philadelphia, 2006.

Abdo Chahin, M.Sc.
Universität der Bundeswehr München, LRT3
Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany
Email: abdo.chahin@unibw.de