

# Investigating Elementary Design Methods

S. Zier, N. Reiss, A. Bohn, H. Birkhofer  
*zier@pmd.tu-darmstadt.de*

## Abstract

The aim of investigating elementary methods is to increase the acceptance of methodical working in practice by developing a systematic approach for the deduction, description and improvement of design methods. The purpose of this paper is to give an overview about the goals and modularization methods of design methods with the Genome Approach.

**Keywords:** *elementary design methods, adaption of design methods*

## Introduction

In the field of product development, there are many methods that are intended to carry out the product development process more effectively and more efficiently. The definition of design methods used here is based on the fact, that periodically occurring sub-processes can be marked-off and standardized for the purpose of rationalization. Each designer uses an individual mix of methods. S/he selects the most suitable design methods and method descriptions depending on several factors, e.g. user skills, infrastructure and working aids [10].

Examples of method descriptions are presented in well known references such as [7], [6] or [9] who assign design methods to certain design phases of product development, or rate them to special access items. In addition, a large number of new methods are introduced in various publications and at significant conferences.

It seems reasonable to assume, that these are not all new methods but rather derivatives of methods that are already known. However, the descriptions often vary regarding the kind and length. Quite often, the descriptions are insufficient for the application of methods in design projects. The kind of description differs in structure and content. The same circumstances or facts are partly described in various terms and in different sequences [1].

In order to clarify this, different method descriptions can be compared with each other with the help of the Genome Approach.

In this paper, the Genome Approach [3] is used to model a certain number of method descriptions of the same method as so-called genomes. Genomes are directed graphs, depicting method sequences as a series of so-called elementary methods. This idea is driven by the vision that the above-mentioned myriad of method descriptions is referable to a limited number of elementary methods. Thus, development and construction methods can be described as formally/ objectively as possible to assess them in a neutral way. With this approach, it is now possible to obtain clarity about basic processes in single methods as well as to show the actual substantial core of each method. These insights allow a combination of elementary methods to applicable product development methods and to create specific sequences of elementary methods according to a specific task and design situation.

Furthermore, the approach serves as a tool to summarize method descriptions objectively. In that way complete method descriptions emerge (e.g. from the "Requirement List") which are allowed to make a claim to universal validity. Providing complete method descriptions of methods is an improvement for the method user. The emerging modular structure of the method genome simplifies the method adaption. Examples will show the application of this approach later in this contribution.

In total, the approach aims to "clean up" the world of design methods, which seemed to be a most demanding, even utopian goal regarding previous attempts of various authors [1].

## Analysing Method

The so-called Genome Approach was introduced to define elementary design methods.

For method users, genomes are clear and complete guidelines. With some practice, the experienced method user manages to compose methods appropriate to the situation from recurring method modules. In method application itself, an easy and unambiguous overview of the required inputs and created outputs represents precious information. The extended Genome Approach uses only two different kinds of elements (Objects and Values) and three different operations (List, Assign, Connect) to model any kind of method descriptions. See table 1.

The linking of two elements with operations resulting in a scalar, vector or matrix is called elementary method. Thereby close attention has to be paid to the operations “assign” and “connect”.

Table 1 does not show that with the help of these operations also various elements (e.g. objects and values) can be assigned to or connected with each other. In table 2 the allocations in the operation “assign” are displayed. The same applies the operation “connect”.

Table 1 Codes for Elementary Design Methods

Operations Elements	List	Assign	Connect
Object	5.1	5-5.2	5-5.3
Value	7.1	7-7.2	7-7.3

Table 2 In Detail: Assign

Assign (X.2)	Object	Value
Object	5-5	5-7
Value	7-5	7-7

In order to model each of the method descriptions as a genome the following modelling-procedure must be carried out:

### Modeling method descriptions as single genomes:

1. Breaking down the original text of the method description to single steps and assign a serial number
2. Translation the single step description to elementary methods and assign the according elementary code
3. Defining in- and output for each step and check for consistency
4. Defining the system boundary and drawing the genome

As an example a translation of an excerpt of the description „List of Requirements“ according to [5] is shown below:

### **1. Breaking down the original text of the method description to single steps and assign a serial number.**

From the original text: „... start with a requirement specification of a known customer, or distribution/marketing. The content does not just consist of technical specifications but also of target cost, ergonomic and safety specifications and Industrial Design.“ Table 3 can be derived.

The cited parts can be separated in step 1 and 2 of table 3. The description of the identified step itself is shown in column C, the serial number in column A.

Table 3 Transforming a method description from prose to a genome

A) Serial No.	B) Elementary Code	C) Description of step	D) In- and Outputs
1	5.1	Collecting of requirements	In: Sources of requirements (L) Out: Requirements (L)
2	7.1	Listing of requirement types (Fixed claims, ..., wishes)	Out: Requirement type (L)
3	5←7.2	Assigning of requirement types	In: Requirements (L), Requirement type (L) Out: List of requirements (T)
...	...	...	...

## 2. Translation the single step description to elementary methods and assign the according elementary code.

In Table 2, the elementary code is assigned to each identified step in column B. The color coding used in Table 2 is according to the later genome color coding. Here first a list of objects (here: requirements) has to be created. Then different kinds of values (e.g. requirement categories) are listed accordingly and assigned to the corresponding requirement.

## 3. Defining in- and output for each step and check for consistency.

In table 3 the in- and outputs are shown in column D. The letter in brackets specifies the format of the in- and outputs. In this example, there are two kinds of formats: Lists (L) and tables (T). A table emerges from assigning a list to another list or table. Next to the contents, the format can also be used to check the consistency of the method description.

## 4. Defining the system boundary and drawing the genome:

Figure 1 shows the Genome of the method “Requirement List” according to the description of [5]. In order to create this, the information from Table 2 was modeled into a directed graph, which describes the transformation of the mentioned lists and tables to a requirement list.

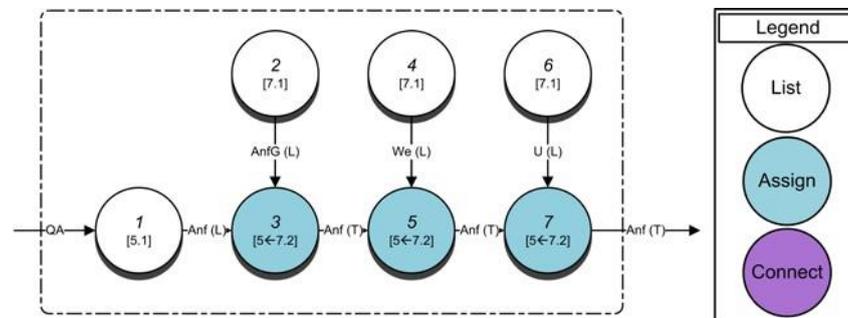


Figure 1 Genome of the method “Requirement List” according to the description of [5]

In this paper, the extended Genome Approach is used to create a “complete” method description of the method “Requirement List” by combining the genomes of several method descriptions from different authors to one overall method genome. In order to generate this, four major steps must be carried out:

1. Modeling each of the method descriptions as a genome as described above
2. Identifying the common core of all single-genomes
3. Combining the single method genomes in one, so-called, author’s-genome
4. Restructuring the author’s genome to a complete-method-genome

By comparing the single method description genomes it is possible to identify the core sequence of the method. The core sequence is a common sequence that appears in every method description of a certain method. Often it is striking, which is the core of a method but it also can be detected by an automated pattern recognition.

After having identified the core of the method, it is possible to combine all single method description genomes in one, so called author’s genome.

The complete-method-genome is developed in an iterative procedure with corrective character from the author’s-genome. The procedure runs off in two steps:

1. Clustering contents concerning together belonging single steps to modules
2. Adding these modules to the already developed complete-method-genome.

## Application Example

In this section, the advantages of method modeling with the help of genomes shall be pointed out in an example.

For that, five different method descriptions of the method “Requirement List” are modeled compliant to the proceeding described above. However, for reasons of the shortage of space only two genomes are shown here. Thus, the method descriptions can be neutrally compared with each other.

However, with the direct comparison of both genomes illustrated here several different and common characteristics of the method descriptions are striking.

Differences exist on the one hand in the number of the described steps. The description of Lindemann uses up to 20 steps, while Ehrlenspiel describes only seven steps. The different number of the steps can be led back on the fact that Lindemann, for example, describes a sorting of the requirements, which Ehrlenspiel renounces. Furthermore, Lindemann describes a more detailed requirement list. Different kinds of values are assigned to the requirements, such as date and responsibility.

However, beside the evident differences at first sight in the method descriptions common characteristics can be identified as well.

In all five modeled methods, requirements are listed first. Then a list with values is assigned element-wise to this list of requirements. There originates a table for demands and accompanying values. Then a list with requirement categories is assigned to this table.

All modeled method descriptions have this sequence in common. Therefore, it is the core of the method. Figure 2 shows how the core genome is identified in the two single genomes of the descriptions of Ehrlenspiel and Lindemann. First, both method descriptions are modeled to genomes according to the above-described procedure. In the second step, the single genomes are compared and common steps are identified.

After the core of the method was identified, the so-called author's genome can be provided.

However, this genome contains all steps that were described by the five different authors. Nevertheless, partially these are still included redundant.

With the modeling of the author's genome there originate methods modules. These modules are concerning the contents separateable method steps. This modular structure simplifies the construction of the whole genome. For reasons of shortage of space, the author's genome of the method "Requirement List" is not shown in this contribution.

It is then possible to form a common genome from the five genomes available, by rearranging the author's genome. Thereby a complete description of the method results, including all modeled method modules. That way the application of methods can be adapted to the given situation, by omitting or adding single method modules. Furthermore, subsequent method steps can be added. See Figure 3.

In the example of the "Requirement List" the core of the method consists of five steps (see grey tinged modules in Figure 3). By carrying out these five steps, the method applier creates a "minimum requirement list", consisting of a table with three columns: requirements, values, and category. These five steps can be clustered in three modules. These modules are mandatory, but the "minimum requirement list" can be extended by six optional modules. Optional steps are separated from the mandatory path with so-called gateways, known from BPMN [8].

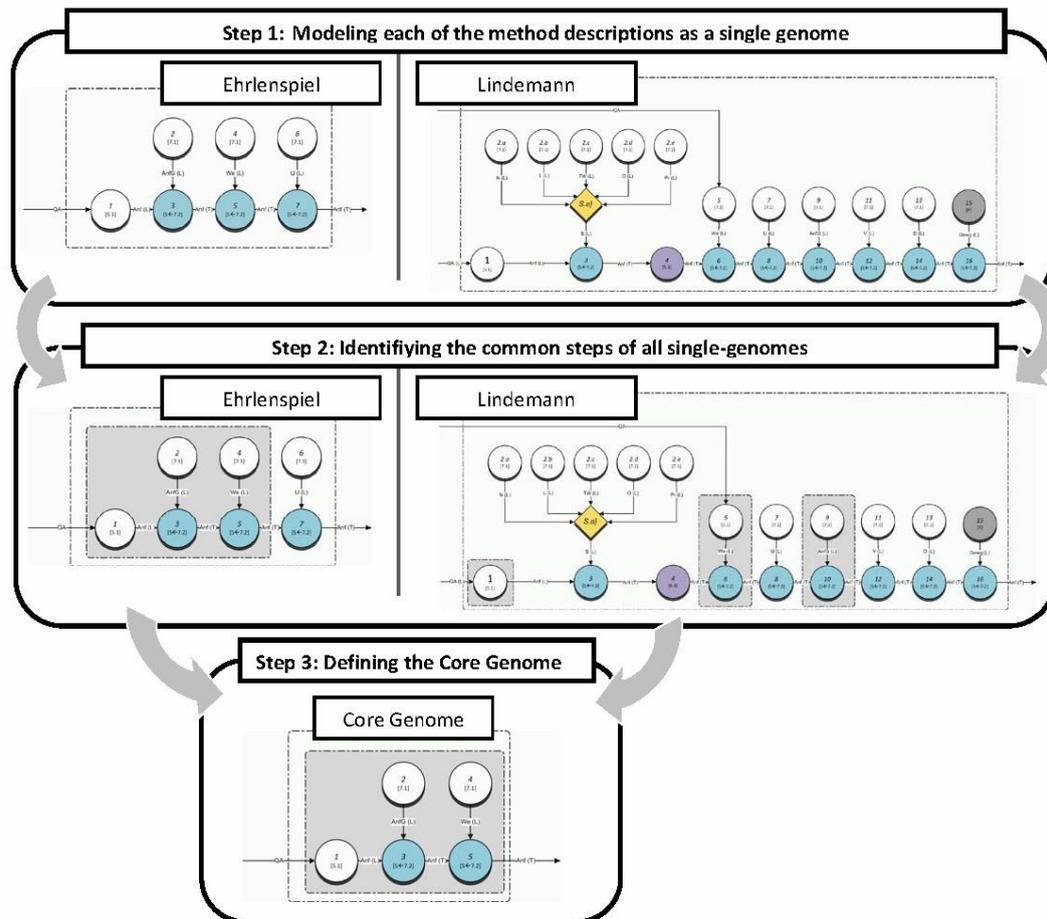


Figure 2 Genome of the method description Requirement List according to Ehrlenspiel

With the help of the gateways, the method applicator can decide whether s/he follows the mandatory path or carries out an optional step in order to improve the quality and number of details of the requirement list. In the example of the method “Requirement List” the six optional modules are:

**1. Selection, preparation and listing of requirements**

This module is described by [7]. Different sources and kinds of requirements are taken in consideration. Furthermore, it is distinguished between implicit and explicit requirements, and explicit requirements are translated with the help of an internal glossary. Result of this module is an improved and distinguished list of requirements.

**2. Sorting of requirements**

This module is described by [6]. The requirements are sorted by means of responsibilities, life-cycle-phase and categorized to technical, economical or organizational requirements.

**3. Checking for feasibility of requirements**

This module is described by [4]. It is checked, whether the listed requirements are feasible concerning certain criteria. Therefore, scientific criteria, juridical criteria, social criteria, ethical criteria, ergonomical criteria, and economical criteria are listed. Afterwards the criteria are assigned to the requirements and non-fitting requirements are whether changed or deleted.

**4. Adding extra information to requirements**

This module is a combination of all single method genomes. Different kinds of extra columns can be added to the requirement table. Examples are sources of the requirements or linkages to other requirements or products.

## 5. Resolving requirement conflicts/ determine degree of requirements

This module is described by [4]. The list of requirements is connected to itself in order to make a check for correlation. Therefore, independent and supporting requirements as well as requirements with goal conflicts become visible. To solve this, the importance of the requirements is rated in a pairwise comparison.

## 6. Releasing of requirement list

The very last module is described by [2]. It is the final control and release of the created requirement list. As mentioned above, this overall genome shown in Figure 3 is a complete summarization of five method descriptions of the method "Requirement List". The method user can choose freely, which modules are suitable for the current situation, product, and available time.

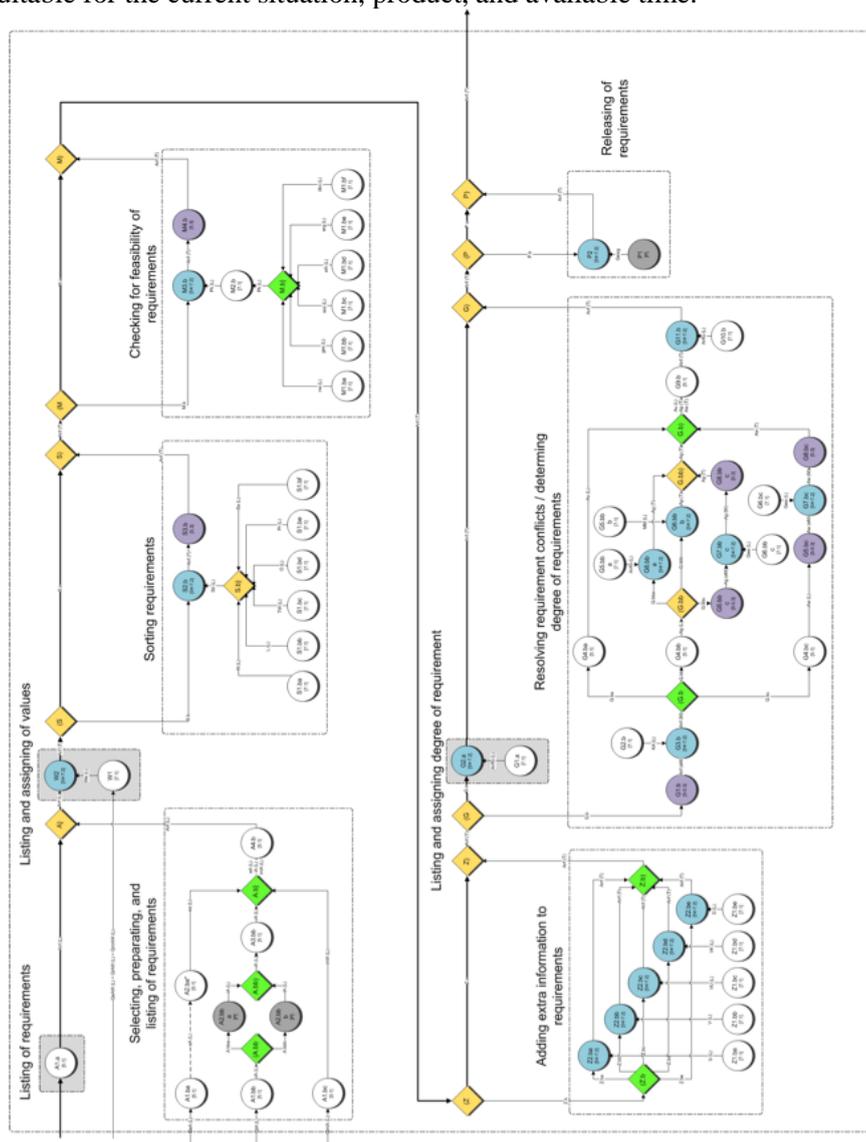


Figure 3 Overall genome of the method "Requirement List" according to Lindemann, Birkhofer, Pahl/ Beitz, Ehrlenspiel, and Breiing

## Automated pattern recognition

In addition to the manual way, an automatic algorithm for pattern recognition was developed. The term "module" refers to a completed functional unit of a method. Modules consist of a sequence of at least two elementary-methods and may occur in different methods. The modularization is used to improve the understanding and the synthesis of methods. The automated pattern recognition can be carried out in two different ways.

## 1. Search for known modules

During the classification, the amount of data is searched for predefined patterns. For this reason, the approach is focused on finding a perfect match of features. Based on expert knowledge and manual analysis, modules are getting identified. After that, the genome gets automatically sought after these modules. Thus it can be checked whether a module is present in the genome

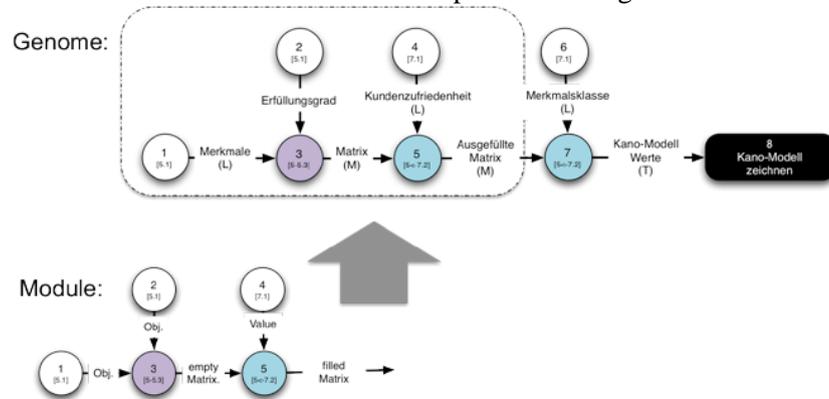


Figure 4 Search for known modules

This algorithm is used to search for the core of a method. When a potential sequence can be identified in all method descriptions of the various authors, it is the core. When a sequence found in several different methods descriptions, it may be a potential module.

## 2. Search for unknown modules of unknown structure

The second analysis method deals with the search for modules, of which neither the precise nature nor the structure is known. Since there is no knowledge of the modules to be scanned and their structure, a heuristic method is needed that allows the gradual approximation to potential modules.

As a starting point for the analysis, the elementary method can be used, which occurs most frequently within one or more methods. Based on the most common elementary method, all of two-elementary sequences are listed, which proceed from this basic method etc..

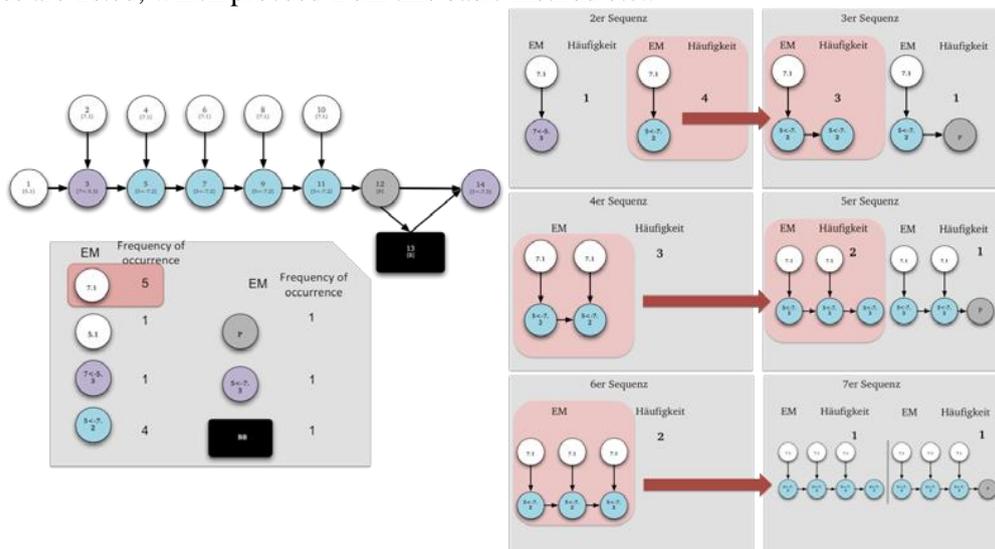


Figure 5 Search for unknown modules of unknown structure

If methods descriptions of different authors are considered, most common sequences can be parts of the core. The sequence must occur here as often as the number of the authors which are considered. If different methods are analyzed, the most common sequences are potential modules.

The listed genome sequences are just a collection of potential modules and not complete modules. It is assumed that sequences that occur frequently, may likely act as modules, as those that occur only rarely. These potential cores and modules are then checked with the first method.

## Conclusions and Further Work

The Genome Approach by means of specific extensions is able to embody a neutral tool for the analysis of methods from different fields. The other objectives of this research project are essentially divided into two main areas.

First, a most complete modeling of existing methods is aspired. Therefore, the elements and operations of the approach have to be extended systematically. The aim is to be able to model any new incoming method as a genome easily in a uniform and complete way. With the help of software (which interprets the genomes as directed graphs), all genomes can then be analyzed and frequently recurring patterns will be recognized. Therewith the actual cores of both, each method and a limited number of recurring method components can be identified. This would clear the diversity of existing method variants.

The second goal is the applicability of the identified method modules. So far this is accomplished by the action leading elements described in this paper (\*-operations, Gateways, Black Boxes). In the future, it is to be fully systematized for example by describing creative processes as a separate genome.

This bears advantages for the application of methods as well as for teaching. Furthermore, it will give a deeper insight in creative thinking of designers.

For method users, genomes are clear and complete guidelines. With some practice, the experienced method user manages to compose methods appropriate to the situation from recurring method modules. In the application itself, an easy and distinct overview about the necessary inputs and created outputs represents precious information.

By knowing that all existing methods rely on a limited number of recurrent patterns, learning methods becomes easier for both, students and practitioners.

Another aspect is the consideration of cognitive skills of designers. The elements (lists) of the Genome Approach are comparable with the operators of the human problem solving. Thus, it could be proved that organization of development methods in elements (lists) and elementary methods generally correspond to a natural human problem solving method. The goal is to support cognitive skills or qualities, which are prerequisite for successful developers, for example, decision making, generalization, association and deductive reasoning.

## References

- [1] Birkhofer, H., Kloberdanz, H., Berger, B. and Sauer, T., Cleaning Up Design Methods: Describing Methods Completely and Standardised. In Marjanovic, D., ed. Proceedings of the 7th International Design Conference, pp17-22 Dubrovnik, 2002.
- [2] Birkhofer, H., Kloberdanz, H., "*Produktinnovation. Lecture Notes 2011*".
- [3] Birkhofer, H., "*Elementary Design Methods and their benefits for research and practice.*" Proceedings of the 16th International Conference on Engineering Design, pp741 Paris, 2007).
- [4] Breiing, A., Knosala, R., "*Bewerten technischer Systeme*", Springer Berlin Heidelberg, 1997.
- [5] Ehrlenspiel, K.; Kiewert, A.; Lindemann, U., "*Kostengünstig Entwickeln und Konstruieren*", Springer München, 2007.
- [6] Lindemann, U., "*Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*". Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [7] Pahl, G. and Beitz, W., "*Engineering Design: A Systematic Approach.*" Springer-Verlag London Limited, London, 2007.
- [8] Silver, B., "*BPMN method and style.*" (Cody cassidy press, Aptos Calif., 2010).
- [9] Ulrich, K.T. and Eppinger, S.D., "*Product design and development*". Irwin McGraw-Hill, Boston Mass. u.a., 2009.
- [10] Zier, S., Birkhofer, H., "*Investigating Elementary Design Methods - A process oriented and model based approach.*" In Marjanovic, D., ed. 11th International DESIGN Conference, pp.87-94Dubrovnik, 2010.