

INVESTIGATING ELEMENTARY DESIGN METHODS- USING THE GENOME APPROACH FOR CREATING COMPLETE AND ADAPTABLE METHOD DESCRIPTIONS

S. Zier, A. Bohn and H. Birkhofer

Keywords: elementary design methods, method adaption

1. 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 [Zier 2010].

Examples of method descriptions are presented in well known references such as [Pahl and Beitz 2007], [Lindemann 2007] or [Ulrich and Eppinger 2009] 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 volume. Quite often, they 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 [Birkhofer 2002].

In order to clarify this, method descriptions can be compared with the help of the Genome Approach. Comparing for instance the descriptions of methods of [Birkhofer 2011], [Pahl and Beitz 2007], [Breiing 1997], [Ehrlenspiel 2007] and [Lindemann 2007], using a relatively distinct method like "Requirement List", it can be seen that the descriptions differ from each other.

If all of these method descriptions are given to a user or student, it does not become clear to her/ him which method description is suitable for the current circumstances of development. In addition, all five descriptions barely give indication of how the method can be adapted to the actual problem at hand.

In this paper, the Genome Approach [Birkhofer 2007] 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 (Figure 1).

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 simultaneously emerging modular structure of the method genome simplifies the method adaption. Examples show the application of this approach later. 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 [Birkhofer 2002].

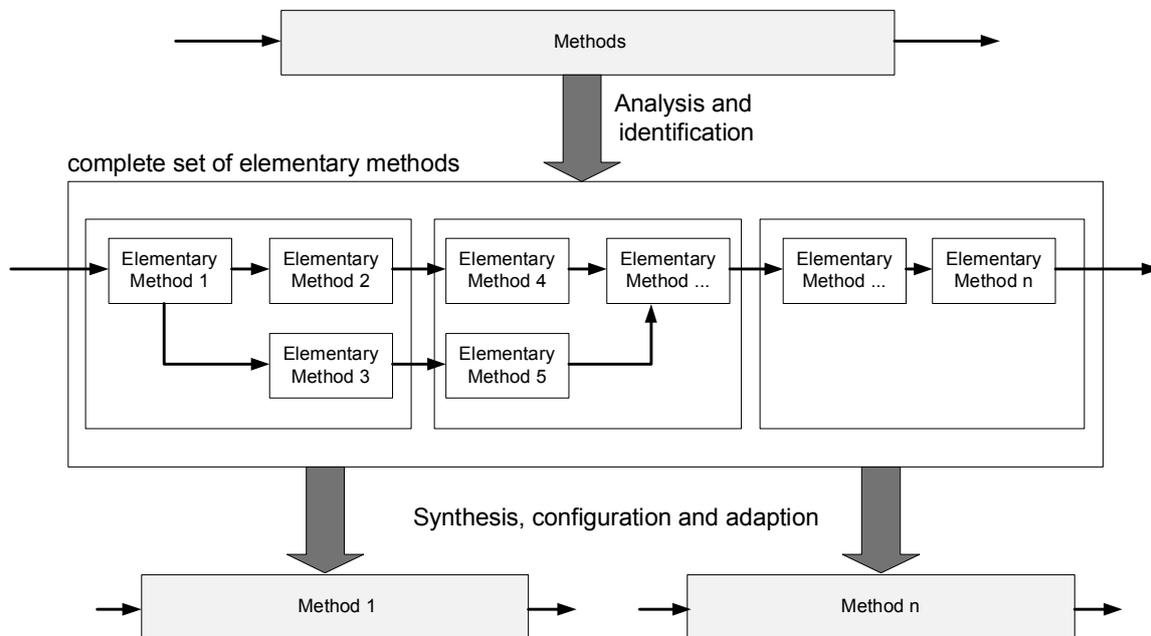


Figure 1. Objective of elementary methods

2. Analyzing method

The so-called Genome Approach was introduced to define elementary design methods. This approach differs substantially from a wide range of other research work that intended to capture the basic nature of design methods or to improve application by giving advices.

The Genome Approach makes use of the often-occurring list (vectors) and matrix structure of design methods (e.g. QFD, FMEA, etc.) to model methods as networks of operations that transfer lists and matrices. Thereby, method descriptions are diagrammed. The generated diagrams remind the beholder of pictures of biological chromosome sets (genomes), which gives the approach its name. 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 methods. 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 (left) Table 2. In detail: assign (right)

Operations Elements	List	Assign	Connect	Assign (X.2)	Object	Value
Object	5.1	5-5.2	5-5.3	Object	5-5	5-7
Value	7.1	7-7.2	7-7.3	Value	7-5	7-7

In order to model each of the method descriptions as a single genome the following subsequent 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 [Ehrlenspiel 2007] 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: „*We 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 2 can be derived.

The cited parts can be separated in step 1 and 2 of Table 2. 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)	B)	C)	D)
	El e m e n t a r y C o d e	Description of step	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)
4	7.1	Listing of values	In: Sources of requirements (L)

			Out: Assigned Values of requirements (L)
5	5←7.2	Assigning of values to list	In: List of requirements (T), values (L) Out: List of requirements (T)
6	7.1	Listing of origin of requirements	Out: Origin of requirements (L)
7	5←7.2	Assigning origins to list	In: List of requirements (T), origin of requirements (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, where each operation is coded with an own color, see legend in Figure 2. In this method description, the operations “list” and “assign” are used alternately. 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 2 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 2 shows the Genome of the method “Requirement List” according to the description of [Ehrlenspiel 2007]. 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.

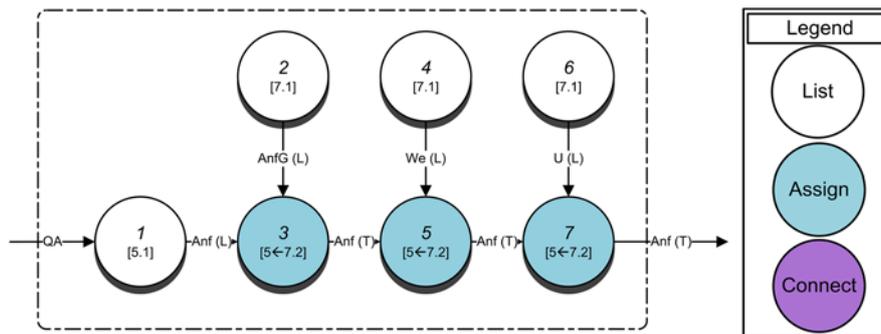


Figure 2. Genome of the method “requirement list” according to the description of [Ehrlenspiel 2007]

The single steps of the genome are comparable with the completion of the well-known requirement form in Figure 3. It is obvious, that the method is about creating a list of objects (here requirements (Step 1)) and extend this list by assigning (Steps 3, 5, 7) corresponding listed values (type (Step 2), value (Step 4), origin (Step 6)). The comparison of Figure 2 and Figure 3 shows, that the created genome is equivalent to the known method description.

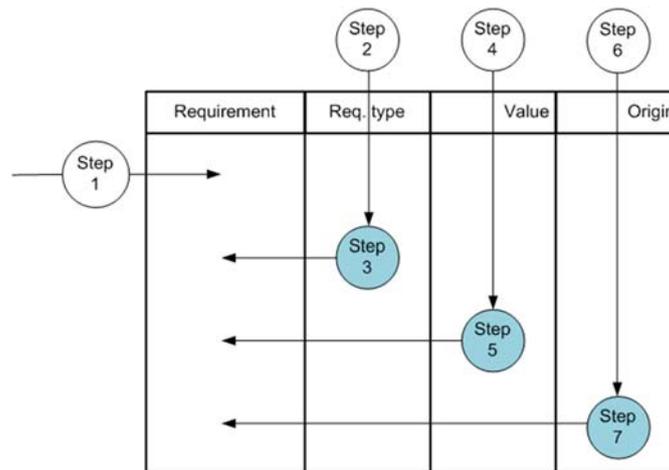


Figure 3. Carrying out the method “requirement list” according to the genome in Figure 2

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 single 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 single method genome 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.

It is suitable to begin with the core of the method. The core of the method is the greatest common subset of all examined single genomes of the method. Besides, the core must not be coherent compelling. Thus it is conceivable, e.g. that the core is interrupted by optional method components.

That means that the complete method genome contains a mandatory part (the core) and optional parts (other clusters).

3. 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 4 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. This step can also run automatically, a corresponding algorithm has been developed, but is not topic of this paper.

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.

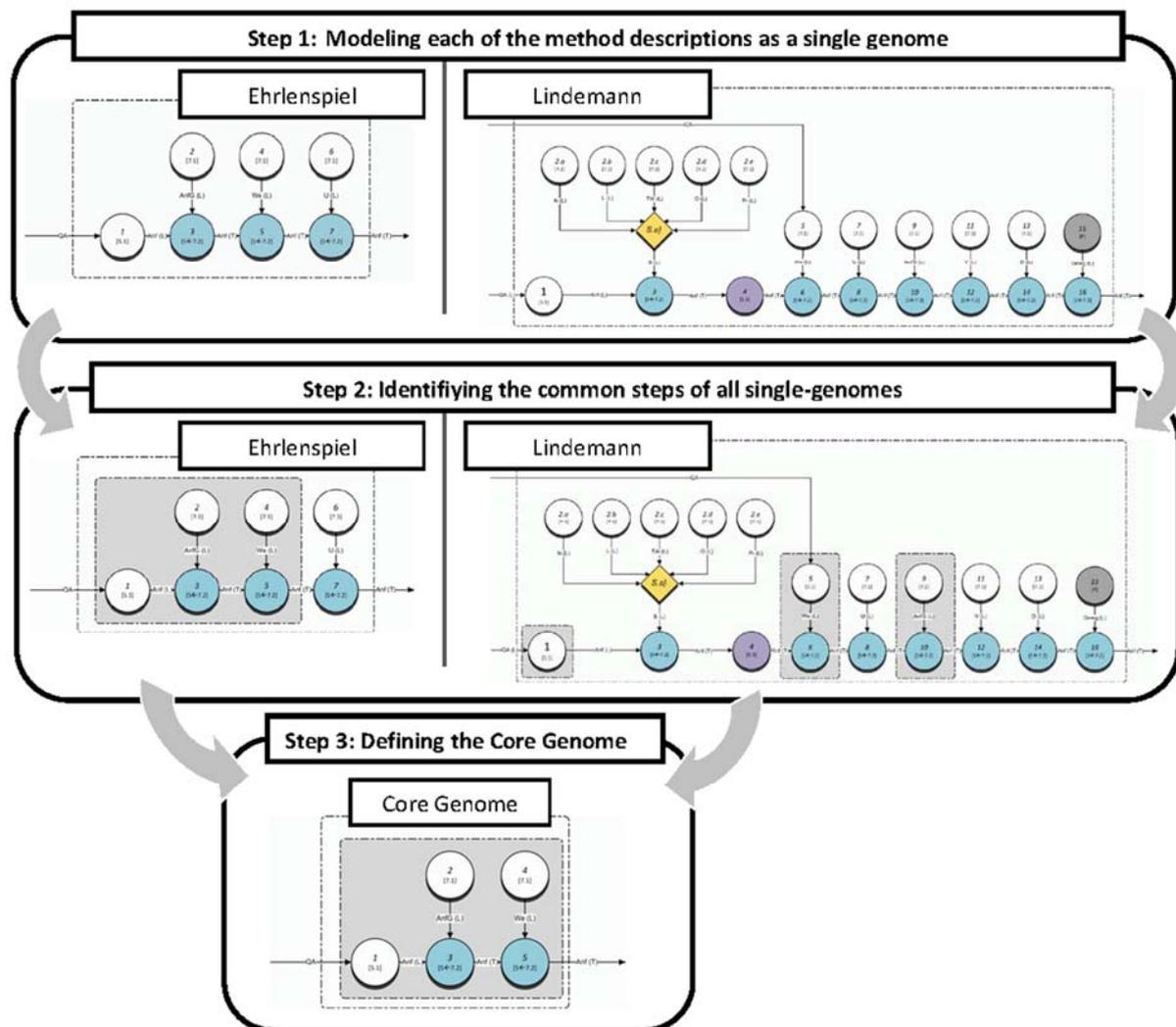


Figure 4. Genome of the method description requirement list according to Ehrlenspiel

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 5.)

This kind of complete method descriptions relieves the method applier with the selection of a suitable method as well as with the adaptation. The user can simply decide which of the optional steps he must carry out for the reaching of his desired aim. However, he also gets to know which steps are mandatory.

In the example of the "Requirement List" the core of the method consists of five steps (see grey tinged modules in Figure 5). 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 [Silver 2010].

In the genome approach, two different kinds of gateways can be distinguished:

- **OR – Gateways:** Here the gateway enables at an input that one or more outputs are further „tracked“. This allows modeling of optional elementary methods or elementary methods only iterated at a fulfilled condition. OR Gateways are displayed green.
- **XOR – Gateways:** Here the gateway enables at an input that one single output is further "tracked". This allows the modeling of case distinctions. XOR Gateways are displayed yellow.

With the help of the gateways, the method applier 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 [Pahl and Beitz 2007]. Here different sources and kinds of requirements are taken in consideration. Furthermore, it is distinguished between implicit and explicit requirements, and explicit requirements (e.g. defined by customer in contracts) 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 [Lindemann 2007]. In this module 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 [Breiing 1997]. Here 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. Here 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 [Breiing 1997]. In this complex module, 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 [Birkhofer 2011]. It is the final control and release of the created requirement list.

As mentioned above, this overall genome shown in Figure 5 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.

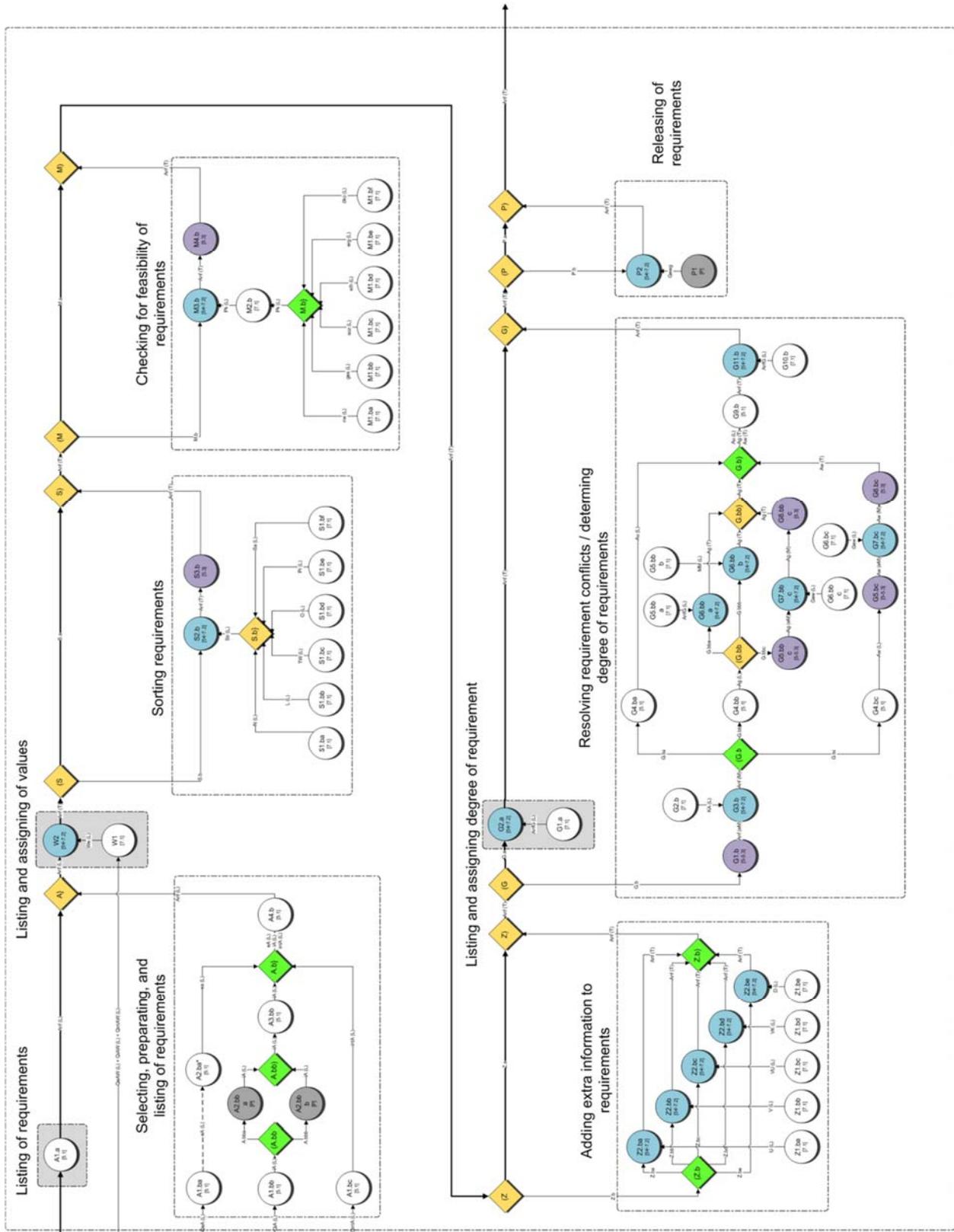


Figure 5. Overall genome of the method "requirement list" combining the single genomes to the method descriptions according to Lindemann, Birkhofer, Pahl and Beitz, Ehrlenspiel, and Breiing

4. Evaluation

The modeling procedure described in this paper was evaluated. Five different people with five different method descriptions carried out the modeling. This led not only to nearly identical single genomes, but also to very similar author's genomes and complete genomes.

5. 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

- Birkhofer, H, Kloberdanz, H., *Produktinnovation. Lecture Notes 2011.*
- Birkhofer, H., *Elementary Design Methods and their benefits for research and practice. Proceedings of the 16th International Conference on Engineering Design, pp741 Paris, 2007.*
- Birkhofer, H., Kloberdanz, H., Berger, B., 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.*
- Breiting, A., Knosala, R., *Bewerten technischer Systeme, Springer Berlin Heidelberg, 1997*
- Ehrlenspiel, K., Kiewert, A.; Lindemann, U., *Kostengünstig Entwickeln und Konstruieren, Springer München, 2007.*
- Lindemann, U., *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2007.*
- Pahl, G., Beitz, W., *Engineering Design: A Systematic Approach. Springer-Verlag London Limited, London, 2007.*
- Silver, B., *BPMN method and style. (Cody cassidy press, Aptos Calif., 2010).*
- Ulrich, K. T., Eppinger, S. D., *Product design and development. Irwin McGraw-Hill, Boston Mass. u.a., 2009.*
- 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.*

Sebastian Zier
Research Associate
Technische Universität Darmstadt, product development and machine elements
Magdalenenstr. 4, Darmstadt, Germany
Telephone: +49 6151 16 2455
Telefax: +49 6151 16 3355
Email: zier@pmd.tu-darmstadt.de
URL: <http://www.pmd.tu-darmstadt.de>