INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

DEVELOPMENT OF AN EVOLUTIONARY BASED DESIGN METHOD

Konstantin Kittel, Sándor Vajna
Otto-von-Guericke University Magdeburg

ABSTRACT
The Auto Genetic Design Theory (ADT) uses analogies between evolution and activities in the design processes to ensure, that the best possible solution can be found, within the given boundaries. These requirements and conditions can also contradict each other and change over time. The ADT interprets the product development as a continuous improvement process. The presence of self-similar activities and the existence of chaotic behavior are characteristics of the ADT.
Based on previous work on the basics of ADT, the actual knowledge should be expanded and deepened; so that they have most activities during product development can be described from an evolutionary perspective. This should be processed by working on the following objectives: a product description from an evolutionary perspective, description of the solution space, development of a process model which describes the processes during an evolutionary product development.

Keywords: Product Development, Biology, Design Method

1. STATE OF RESEARCH
In the field of product development there are numerous methods and process models, which should assist the developer in his work.

For a more deterministic and phase-oriented description of the product development the design accord to the guidelines “VDI 2221” and “VDI 2222”, for their transfer into the Mechatronics the guideline “VDI 2206”, for a model-oriented and axiom based approach the “General Design Theory” and the “Axiomatic Design Theory”. The “Universal Design Theory” as a unifying approach, for a strongly synthesis and analytic approach the “Characteristics Properties Modeling / Property-Driven Development” and for a flexible approach and organization stressed the “Dynamic Product Development”.

TRIZ [23], as a method, which uses contradictions and conflicts to solve problems, for the use of evolutionary algorithms and neural networks the “Evolutionary Algorithms In Design” and the “Integrated Evolutionary Engineering Design”.

In the following, two methods will be presented. The methods were selected, because both use analogies between product development and biological evolution.

1.1. Gene Engineering
Gene Engineering is as a design method to develop new products using genetic algorithms and neural networks. This approach is based on the facts that new products are always based on existing products, i.e. they represent a variation or combination of already known solutions, and (2) the analogy to biological evolution [21].

Gene Engineering consists of five phases [17]:
1. Discharging the virtual chromosomes of an existing product: The product information is stored in a virtual chromosome. Information is split into genetic information (information needed for manufacturing the product) and evolutionary information (information needed for the variation of the virtual chromosome). The evolutionary information can be understood as the possible solution space.

2. Identifying of the virtual genes and creating the virtual genome maps displaying the dependencies of the different genes: Dependencies exist between genes, which influence the same property of the product. Causal networks (in the form of tree diagrams) are created on the basis of the relations between the genes. These networks were used to influence certain characteristics of the product systematically.
3. Deriving mapping networks between virtual genes and product characteristics: Determine associations between genes and product characteristics. The product characteristics to be considered are determined by the designer and / or the customer. The result of this step is a network that displays, which virtual genes influence which product characteristics.
4. Creation of the virtual gene library: This gene library (also named "library of superior genes") based on the statement, that most products use the same effects and principles, of which an engineer usually uses not more than 100.
5. The design process itself is divided into several steps. As a first step, those properties of the (current) product are determined, which deviate the most from the desired goal. For these "defective performance" properties, those virtual genes were detected, which determine these properties. For this step, the network of phase 3 is used to detect the so-called "defective evolved genes". In the last step, the defective genes are exchanged by "superior genes" from the gene library. The selection of these genes is done by a "survival of the fittest" mechanism.

Gene Engineering is an interesting approach to support the product development process, but also raises a few questions:

- The success of Gene Engineering significantly depends on content and quality of the gene library. Solution space and achievable solution quality are limited by the content and quality of the gene library. This can avoid really innovative solutions. The concept of gene engineering is made for the variation of effects and principles, a shape optimisation seems to be impossible.
- Another critical point is the development of mapping network. After [17], the (mandatory required) dependencies between virtual genes and product properties are created by the designer and / or the customer. This presupposes, that these relationships can be seen directly and can be allocated correctly. This should be possible with products with simple connections. Whether a correct mapping of dependencies for complex products is always possible, too, has not been clarified. Even the more complex products, whose connections are not that simple, normally show a greater optimisation potential.

1.2. Evolutionary Integrated Engineering Design

With the Integrated Evolutionary Engineering Design (IEED) FAN, ANDREASEN and others describe an approach [3], which uses a specific chromosome model (based on [4]), to code the characteristics of a product with the objective of reaching a better description the optimisation problem. The main problem within the product development process is the incomplete coding of the product by the designer. Since there is no general approach, the way of coding depends on the designer. This leads to reduced and divergent models, which do not fully reflect the requirements.

To solve the encoding problem, a chromosome model is used. The structure of chromosome is done by modeling a product from the following perspectives:

- Process view: Describes the structure of the possible processes during product usage. These processes are needed for the realisation of the material, energy, and information flows.
- Functional view: Describes the structure of the functions and effects needed for the realisation of the required material, energy, and information flows.
- Organ view: Describes the structure of the functional carriers needed to fulfill the required functions.
- Part of view: Overview of the existing parts and their assembly relationships.

These four views represent the different levels of abstraction for the new product.

The process of IEED starts configuration of the requirements for the new product. Based on the requirement specification, the chromosome model can be "filled". In a first step the material, energy, and information flows are determined. In the next step, the functions and function structures are determined necessary for the realisation. In the last step, the components necessary to fulfil the function are determined.

The creation of an optimised product concept is called "Evolutionary Computation”. Therefore, a four-step cycle consisting of creation, evaluation, reconfiguration, and guidance is passed occurring variants are verified against a fitness function. The optimisation process should not run as an autonomous process, which creates complete result after a certain cycles, but should support the developer.

IEED differs from other approaches mainly through its complex chromosome model, which allows the representation of the solution in various levels of detail. This chromosome contains process-oriented, functional, structural and geometric information, which is much more than within other approaches. According to [3] optimisation is only taking place on the conceptual level. The optimised design is then converted into a physical structure by using a component library. Functions for the optimisation of the physical structure are not provided.

1.3. Some Limitations of "Classical" Design Methods

Basic goals of developing design methods have been standardisation and economisation of procedures in the practice of design as well as the substitution of general problem solving processes by design activities connected with detailed directives of procedure for the designer. These goals determined during decades the educational concepts for the urgently needed engineers.

Design methods like VD I2221 define sequential phases of an individual product solving process. Each phase results in an abstraction respectively a concretion level of the product, which is defined as a technical system, and represents these results in a product model [20]. Especially gradual methods (like requirement lists, function structure, searching for active principles, morphologic boxes for varying and evaluating partial solutions, choosing and combining to complete solutions) are available for seeking solutions. Some researchers tried to represent these methods by algorithms in IT systems, but these attempts have remained unsuccessful so far.

In addition, practice and education showed over the years that design methods have not always brought the wanted effects regarding finding new and innovative product ideas. Some of the reasons for this are:

- On the one hand, design methods are not based enough in empiricism. On the other hand, there are attempts to create a design theory purely from empirical observation that were taken on a very limited number of items (e.g. within one company), of which the findings are generalised.
- The phase models and action schemes of most of the methods are rigidly linear. Discursive iterations, rebounds, and loops are missing.
- Cognition-ergonomic results concerning individual patterns of thinking and acting of product developers are missing.
- The basically rational and gradual forms of notional problem solving strategies impede creativity and innovation and thus limit the space for possible solutions.
- Most design methods are based on the paradigm "divide et impera", i.e. to decompose a given (complex) problem into small elements that can be handled easier. However, the decomposition procedures do not seem to be powerful enough for handling complex design problems: „The ensemble is more than the sum of its different parts“.
- Organisational and economic aspects as well as management aspects for co-operative product development are missing.
- Design methods can only be transferred on a quite high abstraction level to other fields of development like mechatronics or software.

Basic criticism is that design methods are founded mainly on descriptive observations and originate from a time when the focal point of development was creating mechanic assemblies for the investment goods market, which were produced in quite great series, and which had not to be too much individual.

Neither product individualisation (and thus smaller series) was not in the focus nor the emerging market of consumer goods (with much more focus e.g. on industrial design and handling features).

A special problem is the prescriptive, linear-sequential description of design procedures. These procedures impede individual thinking and acting by e.g. heuristic or evolutionary characteristics and therefore, disturb or even hinder creativity and innovation.

By separating design problems to single parts and afterwards finding partial solutions, a main solution is generated that is often unbalanced because decisive product characteristics have not been adequately represented in the creation process. Often, ergonomic, industrial design, and ecological product requirements are simply „forgotten“[22].

Another problem is the adaptation of design methods to the individual. Increasing product complexity as well as the increasingly spatial separation between development and design, often even crossing borders, results in a separated product development in interdisciplinary teams. These new forms of organisation that are supported by new communication and product data management tools are only represented in design methods with an integration approach, like e.g. Integrated Product Development. Finally, product complexity itself as the decisive element in developing methods and procedures is subject to rapid change. New requirements of product development processes excite consideration of new forms of support for design practice and serve at the same time as approach for making an adequate mark in education.

2. OWN PREVIOUS WORK

From the own work, the “Low Sequence Design” and the “Featurebased Product Modelling” will be introduced briefly. Parts of these results were used to formulate the ADT.

2.1. Low Sequence Design

The idea of the approach of the “Low Sequence Design” is that in addition to the description of process also the results of the single work steps were saved into so-called sub-models. These sub-models are connected to each other through some form of inheritance of data and information.

During the product development process, the completeness of the description of the product increases, while the sub-models are generated successive as results of the individual steps. On the one side, the content of a sub-model arises from the contents of previously generated sub-models (due to inheritance, rules, boundary conditions), but simultaneously restricts the contents of the following sub-models. The degrees of freedom for the design for the downstream sub models are also determined by the degree of fulfilment of the requirements and the relevant boundary conditions.

The sum of all sub-models, which form the product model, do not only represent the result of the design process, they also represent the development history of the various phases of design ("evolutionary" information). The dependencies between the sub-models represent the design intent.

For each sub-model can be shown, that:

- Catalogs with predefined solution elements of different complexity (for example, construction Catalogs [7]) and systems for combinatorics (such as the morphological box [18]) are able to generate new content as well as the resulting relationships reproducibly.
- There are tools, which can extract the content and the relations of new sub-models out of existing sub-models and also pretend successive content and relations based on existing information with sufficient accuracy.

Those dependencies (existing for both directions) of the content and the relations, make it possible to understand or predict these contents and relations. That is true not only for directly adjacent sub models, but also for more distanced sub-models. Because of this fact, the development of any product can be started or continued with any activity on any sub-model, at any time.

2.2. Featurebased Product Modelling

In the context of the FEMEX initiative and the creation of the VDI guideline 2218 "Feature based modeling" [9] research on feature based product modeling [14] [19] was done. The main concept of the feature based product modelling is to create a use-oriented view of a unified information model that contains all data and information of a product, which were produced during the product life cycle. Because this information is not only geometric data and simple information it is called extended feature. They form a separate class of information objects and integration objects with the following characteristics [9]:

- Extended features are information technology elements, which represent areas of special interest of a product, and thus enables them to save results from different stages of product development.
- They consist by the sum of properties of a product, which contain the relevant properties, their values and their relations.
- Each extended features represents a specific view on the product description, which is related to certain property classes and certain phases of the product life-cycle.
- An extended feature contains only in certain relevant characteristics of the product. A division into property classes and phases of the product live cycle is necessary for an accurate definition and classification of the different feature types.
- Extended can contain properties from different property classes.

Due to the structuring of the product life cycle in application-dependent phases and the subdivision of the product description into property classes enables this feature definition to create a consistent description of the results of all activities.

An exemplary realisation of the second approach was the design system IKA, which allows a continuous computer support for the apparatus engineering. The focus was on developing a universal approach and application-independent tools. This tools makes it possible to design own programs for the neutral dimensioning of apparatus elements and integrate them in different CAD systems [12] [13].

3. ACTUAL DEVELOPMENT STATUS OF AUTOGENETIC DESIGN THEORY

The ADT is not a bionic based approach, but rather seeks to transfer procedures from biological evolution to accomplish a broad description of the product development and its processes, requirements, boundary conditions and objects (including their properties).

With previous research, a description of the basic properties of ADT is possible as well as a limitation of the research area. In the following the main basics and a possible process of the ADT shall be presented. It needs to be pointed out, that the ADT is still under development. That's why the whole theory is not tested practically yet. A software tool [NOA – Natural Optimisation Algorithm] was developed, which simulates some of the ideas of the ADT. This software tool was used, to optimize few different technical products ([24]).

3.1. Basics

One main characteristic of the natural evolution (with the principle of trial and error) is continuous development and permanent adaption of individuals with minimal use of resources on a changing target, which is dynamic because of changing requirements, conditions, and constraints. This suggests that evolution can be described as a kind of optimisation.

In the evolution, the method for creating individuals is determined by evolutionary operators. These are replication, mutation, recombination, and selection. Especially the mutation (whose outcome and occurrence is not predictable) leads to new ideas, insights or unexpected solutions. The recombination is the combination of different already known principles to create a new solution. The selection is the operator for selecting appropriate solutions from a given set of alternatives.

Main weakness of the ADT is a result of its evolutionary based approach. For reaching a certain quality level, a high number of individuals need to be evaluated. Compared with other methods, the number of solutions, which need to be evaluated, is much higher.

The analysis of the product development from an evolutionary perspective leads to the following insights:

- In every phase of product development process, various alternatives are developed and compared. The alternatives are in competition with each other, because only the best were selected for further processing.
- The processes of searching, evaluating, selecting and combining are also typical approaches of the natural evolution.
- Regardless of the phase of product development or complexity level, always a similar pattern of activities is used to generate new solutions. It is comparable with the TOTE-Scheme [14] [15]. Self-similarity can be found at all levels of complexity of the product as well as in all stages of emerging product [16].
- According to the chaos theory attract small changes or disruptions in the system can cause an unpredictable system behavior according to [17]. The fact that the result of a product, for example, can not be predicted definitely (because of the influence of the creativity of the product developer), led to the assumption, that the product development process also contains elements of a chaotic system or shows a chaotic behavior in some aspects.

An evolutionary product development can be describe as a complex dynamic network over several levels of complexity to present state of research, The development is characterised by the evolutionary operators replication, recombination, mutation and selection at all levels of complexity. At all levels self-similar actions take place. Thereby, those properties where handed over to the successive solutions, which satisfy the prevailing requirements at a particular point of time the most accurate. This so-called autogenesis is recognisable in the creation of any (partial) solution, because any solution must go through this process of self-development. The goal of the optimization itself is changeable, just like the (usually dynamic) boundary conditions.

3.2. Process of the ADT

At the beginning of the process of the ADT, a dynamically changeable solution space is spanned within which the evolution takes place. This solution space is defined by the current requirements, conditions, and constraints. If an external event (such as changing a requirement) happens during the evolution, there will be a change in the solution space, whereby other possibilities for the evolution can arise. It is therefore necessary to extend the search again on the whole "new" solution space in order to get those solutions that consider all actual aspects (Figure 1).

Within figure 1 the solution area is depicted as a rectangle (simplified representation). The increasing level of performance of the emerging solutions is represented by the change of colours from red to green. Starting a certain set of requirements, conditions, and constraints defines the solution space. The evolution is creating appropriate solutions, represented by the coloured narrowing area within the solution area:

- At the first point in time, a reduction of the requirements and conditions happens. This leads to an enlarged solution area. The search has to be expanded to the full and new solution area again, because new possibilities for solutions arose.
- At the second point in time, there is an increase within the requirements and conditions. As a consequence, the solution area is downsized, because there are fewer possibilities than before. Again, the search has to be expanded to the full and new solution area.
- All subsequent activities are similar until those solutions are created that fulfil best the actual requirements, conditions and constraints. The results will consist of several equivalent but not similar solutions.

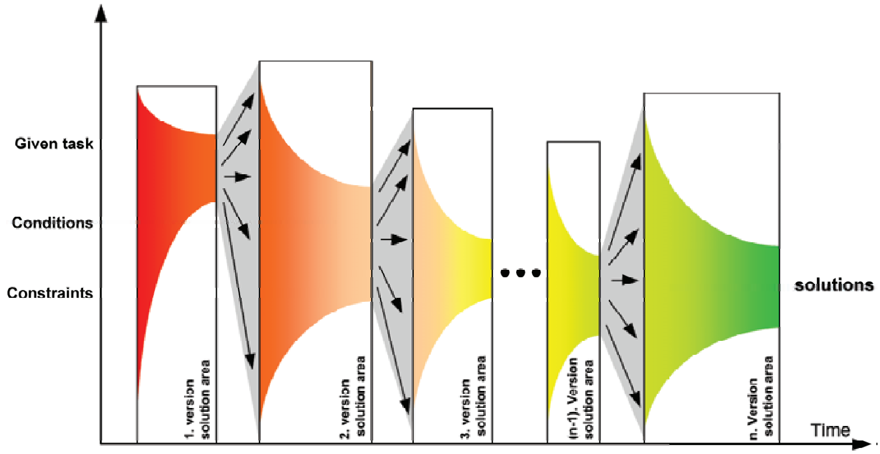

Figure 1: Different occurrences of the solution area during an evolution

The main difference between the ADT and the funnel approach of "classical" design methods is the permanent adaptation of the solution area and the subsequent expansion of the search to the new complete solution area, which consequently try to downsize the solution area to a (single) solution.

4. FURTHER DEVELOPMENT OF ADT

The previous work on the ADT builds the necessary basis for its further development. This includes the basic theories of AKT (optimisation approach, analogy to the natural evolution, self-similarity, chaotic behavior) and the (specific) design and implementation of software engineering tools. In our further work, the basis for the ADT will be detailed and broadened in order to describe more product development activities from the evolutionary perspective and also enlarge the support trough specific computer tools.

In the further work there are four main topics we will work on.

4.1. Product description from an evolutionary view

The goal is, to describe a product from an evolutionary perspective similar to the description of a chromosome in natural evolution. The structuring of content (genes) of the chromosomes shall be based on the approaches of the Low Sequence Design and the Feature based Product modeling (for the internal computer realisation).

4.2. Solution space description from an evolutionary view

The description of the solution space of a product development from evolutionary perspective. It is necessary for the ADT to find appropriate strategies and approaches for describing and searching the solution space, which allow representing dynamic requirements, conditions, and constraints.

It is necessary to examine whether there are different views on the solution space (such as geometry, design, ergonomics …), which show only a part of the product information. If these different views on the solution space exist, it is important to determine what views exist at which point of time. It is also important to examine whether transformation rules between different views can be formulated. In this context, the question arises, what dependencies exist between the different views and how these dependencies can be described in an appropriate form.

4.3. A method for determining the current product progress

It should also be investigated whether a "target chromosome" can be defined (based on the requirements and conditions), which the emerging solutions can converge against. With the help of such a method ("Design Spell Checker"), ideally a permanent monitoring of product development progress would be possible (based on requirements and conditions).

4.4. Procedure model for the ADT

To merge the described activities and with reference to the own work on the procedure model (e.g. [18] [19] [20]), a procedure model for the ADT shall formulated. This model will differ from other models by using methods similar to the natural evolution. This model is an organisational, methodological and technical reference for the evolutionary product development process and shall also represent the current of development in all phases.

5. DISCUSSION

The actual development of ADT is far from being ready. At present, basic procedures and first approaches for the implementation are defined. One practical implementation of aspects of the ADT is the optimisation tool NOA (Natural Optimisation Algorithm). NOA is a genetic algorithm based optimisation software with a universal design which makes it suitable to solve nearly all optimisation problems (if the object which should be optimised can be described parametrical) on different kind of computer hardware.

An important point is the description of the product model, since this is the basis on which all methods of ADT will work. It is necessary to answer the question whether it is possible to find an appropriate evolution-similar description form. It also need to be clarified, how far it is useful to orientate on the biological "product model", consisting of chromosomes and genes. First ideas from a model, consisting of multiple dependent views, suggest that the biological chromosome model will only be the base.

Another open question is how the creation of new products can be described by appropriate methods and procedures. With the current methods of ADT and on the genetic algorithms-based software tool NOA, "only" the further development / optimisation of existing products is possible. The creation of new products, which is equal with emergence of new species in biological evolution, is therefore an important research field. Initial ideas for the transfer of methods of biological macroevolution (macroevolution is responsible for the creation of new species) in the field of product development already exist.

6. SUMMARY

With this paper, the state of research on the ADT should be presented. The selection of already existing methods in the field of evolution-similar design methods shows potential this analogy.

It was shown that the implementation of the dynamics of product development is a priority in the development of the ADT. This is a difference between the ADT and the “classical” methods, which can not respond adequately on changes in requirements.

A key idea of the ADT is the transfer of methods of the natural evolution in the field of product development, whereas bionics tries to transfer biological solutions to solve technical problems. With this approach, the genesis of individual products is knowingly regarded as evolution that can be described as a self-similar iterative procedure with respective driving forces from evolution in the individual design phases. The ADT shall model all procedures and activities in the design process with the evolutionary operators (for example recombination, mutation and selection) [21] [16] [22].

REFERENCES

[1] Chen, K.Z., Feng, X.A.: A Framework of the Genetic-Engineering-Based Design Theory and Methodology for Product Innovation. ICED03, Research for Practice – Innovation in Products, Processes, and Organisations. In: Norell, M. (Editor): ICED03, Research for Practice – Innovation in Products, Processes and Organisations. Vortrag 1745, Design Society 31, Stockholm 2003, Abstract S. 671

[2] Chen, K.Z., Feng, X.A.: Gene Engineering, Product Innovation, Design Theory and Methodology. Journal of Engineering Design 2006

[3] Fan, Z., Andreasen, M.M., Wang, J., Goodman, E., Hein, L.: Towards an evolvable chromosome model for interactive computer design support. In Samuel, A (editor): Proceedings of the 15th International Conference on Engineering Design, Paper 357.46. Melbourne 2005

[4] Andreasen, M. M.: Designing on a Designer's Workbench (DWB). In: Hubka, V.: Proceedings of the 9th WDK Workshop Rigi 1992

[5] Verein Deutscher Ingenieure: Richtlinie VDI 2221, Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. VDI-Verlag Düsseldorf 1993

[6] Gatzky, T.; Naumann, T.; Industriedesign und Arbeitswissenschaft im Integrierten Produktentwicklungsprozess. Proceeedings Kolloquium 40 Jahre Arbeitswissenschaft an der Otto-von-Guericke-Universität Magdeburg, 2004

[7] Roth, K.: Konstruieren mit Konstruktionskatalogen (Band I/II). Springer-Verlag Heidelberg Berlin 1994

[8] Zwicky, F.: Entdecken, Erfinden, Forschen im morphologischen Weltbild, Droemer Knaur Verlag München Zürich 1966

[9] Verein Deutscher Ingenieure: Richtlinie VDI 2218, Feature-Technologie. VDI-Verlag Düsseldorf 2003

[10] Ovtcharova, J., Weber, Ch., Vajna, S., Müller, U.: Neue Perspektiven für die Feature-basierte Modellierung. VDI-Z 140(1997)3, S. 34-37

[11] Vajna, S., Podehl, G.: Durchgängige Produktmodellierung mit Features. CAD-CAM Report 17(1998)3, S. 48-53

[12] Strohmeier, K., Vajna, S.: Koordinierter Schlußbericht zum Forschungsvorhaben AIF 12055 B1/B2, Optimierung der Apparatekonstruktion durch integrierte Rechnerunterstützung. DECHEMA Frankfurt 2001

[13] Zirkel, M., Pilhar, S., Strohmeier, K., Vajna, S.: Optimization of Apparatus Design by Integrated Computational Support. Chemical Engineering & Technology 25(2002)8, S. 789-793

[14] Miller, G.A., Galanter, E., Pribram, K.H.: Strategien des Handelns. Pläne und Strukturen des Verhaltens (2. Auflage). Klett-Cotta Stuttgart 1991

[15] Ehrlenspiel, K.: Integrierte Produktentwicklung (dritte aktualisierte Auflage). Carl Hanser Verlag München 2007

[16] Wegner, B.: Autogenetische Konstruktionstheorie – ein Beitrag für eine erweiterte Konstruktionstheorie auf der Basis Evolutionärer Algorithmen. Dissertation Universität Magdeburg 1999

[17] Briggs, J., Peat, F. D.: Die Entdeckung des Chaos. Carl-Hanser Verlag München 1990

[18] Freisleben, D.: Gestaltung und Optimierung von Produktentwicklungsprozessen mit einem wissensbasierten Vorgehensmodell. Dissertation Universität Magdeburg 2001

[19] Naumann, T., Speck, H.-J., Vajna, S., vom Ende, A.: Relationship between Process and Product Structures - A New and Flexible Approach for an Integrated Dynamic Process Management. In: Stelzer, R., Fichtner, D: Proceedings of CAD 2002 – Corporate Engineering Research, Technische Universität Dresden 2002, Seite 65-76

[20] Clement, S.: Erweiterung und Verifikation der Autogenetischen Konstruktionstheorie mit Hilfe einer evolutionsbasierten und systematisch-opportunistischen Vorgehensweise, Dissertation Universität Magdeburg 2006

[21] Bercsey, T., Vajna, S.: Autogenetischer Ansatz für die Konstruktionstheorie. Beitrag zur vollständigen Beschreibung des Konstruktionsprozesses. CAD-CAM Report 13(1994)2, S. 66-71, und CAD-CAM Report 13(1994)3, S. 98-105

[22] Clement, S., Jordan, A., Vajna, S.: The Autogenetic Design Theory – an Evolutionary View of the Design. In: Norell, M. (Editor): ICED03, Research for Practice – Innovation in Products, Processes and Organisations. Vortrag 1745, Design Society 31, Stockholm 2003, Abstract S. 689

[23] Altshuller, G.: 40 Principles – TRIZ Keys to Technical Innovation (translated and edited by L. Shulyak, S. Rodman). Technical Innovation Center Worcester MA (USA) 2003

[24] Vajna, S., Clement, S., Jordan, A., Bercsey, T.: The Autogenetic Design Theory: an evolutionary view of the design process. Journal of Engineering Design 16(2005)4. S. 423 – 440

Contact: Konstantin Kittel
Otto-von-Guericke-University Magdeburg
Chair for Information Technologies in Mechanical Engineering
Universitätsplatz 2
39106, Magdeburg
Germany
Tel +49 - 391 - 67 - 18093
Fax +49 - 391 - 67 - 11167
E-mail konstantin.kittel@masch-bau.uni-magdeburg.de
URL lmi.uni-magdeburg.de

Konstantin is a PhD at the Chair for Information Technologies in Mechanical Engineering at the Otto-von-Guericke-University Magdeburg. He is giving lectures on computer aided design. He is working on the field of design research and computer aided optimisation especially with genetic algorithms.