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

MORPHIX: AN EVOLUTIONARY WAY TO SUPPORT CONCEPTUAL DESIGN

A. Wünsch, F. Pilz and S. Vajna

Keywords: conceptual design, autogenetic design theory, genetic algorithm, integrated design engineering

1. Introduction

In the development of innovative products, conceptual design takes an important role in the product development process. Final quality and success of a product are significantly influenced by this phase. Since it was introduced by Zwicky in 1966, the morphological box is a commonly used method to generate concepts at different levels of abstraction. By dividing the overall product function into sub functions and by finding different possible solutions to each sub function the totality of all conceivable solutions to a problem can be determined and the optimal solution can be found [Zwicky 1966]. To find the optimal solution the designer needs to evaluate the concepts based on predefined evaluation criteria [Großklaus 2014].

One advantage of this method is the creation and evaluation of a variety of concepts in an early phase of the development of the product. This is done on a high level of abstraction. So the designer can detect early, which solutions satisfy the subsequent function of the product best. The objective is to identify the global optimum that fulfils the criteria best. Due to the high level of abstraction, this method is applicable to any product development process.

The capability of the morphological box to show the totality of all conceivable solutions for a problem is its strength and its weakness at the same time. Even a small morphological box that consists of five sub functions and five different solutions to each sub function leads to 3125 possible solutions for the overall problem. The designer can't evaluate all possible solutions. He has to focus on certain solutions and therefore ventures to miss a good solution.

To reduce the risk of missing a good solution we developed the Morphix framework, a system to support conceptual design by combining the morphological box and the Autogenetic Design Theory [Kittel et al. 2011a] by using NOA, a framework that contains several Genetic Algorithm objects to support the product development process. Genetic Algorithms are very suitable for supporting conceptional design, because they can deal with different kinds of parameters and the objective function doesn't have to be continuous. It also examines the full solution space, which yields to a high probability to find the global optimum.

The structure of this paper is as follows. Firstly, we present the Autogenetic Design Theory and its key features including theoretical background, present research, and application.

Furthermore we describe the development of Morphix which contains the graphical user interface, the usage scenario, the data interfaces to NOA, and the implemented product attributes of Integrated Design Engineering [Vajna et al. 2014], to support the evaluation process. We show the functionality of Morphix in three case studies of real product development examples. Finally the results are discussed and both a conclusion and an outlook are presented.

2. The Autogenetic Design Theory

The Autogenetic Design Theory (ADT) is an approach that transfers procedures from natural evolution to accomplish a broad description of product development and its processes, requirements, boundary conditions, and objects (including their properties) ([Bercsey and Vajna 1994], [Vajna et al. 2005], [Kittel et al. 2011a]). The ADT is not another variety of Bionics. In Bionics, results of the biological evolution are transferred to technical products, e.g. the structure of trees or plants. The main thesis of the ADT is that biological evolution and the process of developing products are quite similar. One main characteristic of biological evolution is the continuous development and the permanent adaptation of individuals with minimal use of resources on changing targets, which are dynamic because of changing requirements, conditions, and constraints. This characteristic is based on the underlying evolution principle of trial and error. The changing requirements, conditions, and constraints can contradict each other and can change over time. This suggests that evolution can be described as a continuous improvement process or as a kind of optimisation.

In evolution, the methods for creating individuals are determined by evolutionary operators, which are selection, recombination, and mutation. Selection is the operator for choosing appropriate solutions from a given set of alternatives. Recombination is the combination of, usually two, different already known principles to create a new solution. Especially, mutation leads to new ideas, insights, or unexpected solutions, because both occurrence and outcome are not predictable.

An evolutionary product development can be described as a complex dynamic network over several levels of product and process complexity. This development is characterised by the evolutionary operators at all levels of complexity. Thereby, only those properties are handed over to successive solutions that satisfy the prevailing requirements the most accurate at a particular point of time. This so-called autogenesis (self-development) is recognizable in the creation of any (partial) solution, because any solution must pass this process.

An algorithm that simulates the biological evolution process as mentioned before is the Genetic Algorithm (GA). The basic scheme of a GA is shown in Figure 1.

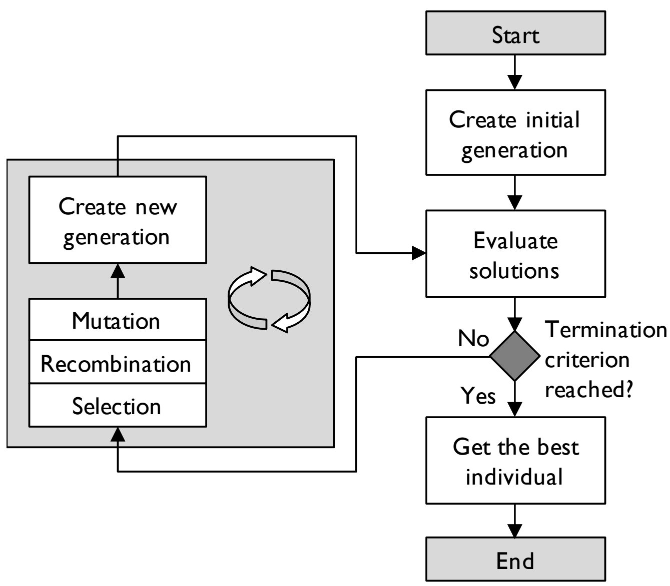

Figure 1. Basic scheme of a genetic algorithm [Pohlheim 2000]

The GA starts with an initialisation phase, followed by an iteration loop of evaluation, which takes on until a termination criterion is reached. The initial generation is a randomly generated set of individuals with different features. A so-called fitness value is used to evaluate each individual of the population. This fitness value represents the performance of the individual evaluated, taking into account the given design or optimisation problem and its requirements. The greater the fitness value compared to other individuals, the higher is its chance to be selected for further evolution.

In general, the goal of the selection is to choose those individuals from the actual generation that will create the next generation. The chance to be selected results from the survivability of each individual. Different approaches exist for the selection of individuals, e.g. uniform selection, stochastic universal sampling, steady state selection or roulette selection. After an individual was selected the GA creates the new generation by using genetic operators to generate better or "fitter" individuals by increasing the distribution of especially good genes. After the selection the genetic operators recombination (which is also called "crossover") and mutation are applied to the selected individuals (Figure 2).

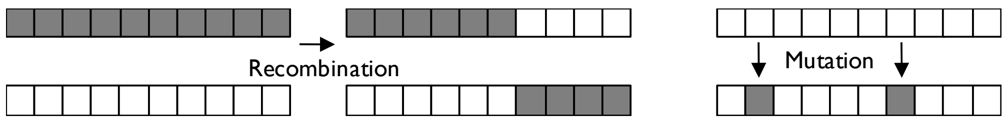

Figure 2. Genetic operators recombination and mutation [Gerdes et al. 2004]

A new generation of solutions, which has a higher rating as the solutions in the generation before, is created after every evaluation loop [Gerdes et al. 2004]. The process of evaluation and selection continues until a termination criterion is reached. When a sufficiently good generation has been reached, this criterion terminates the process.

There are two kinds of termination criteria: direct and derived criteria. Direct criteria are reached at a steady value, such as a predefined number of generations. Also a nearly good fitness-value in comparison to the global optimum, if known, can be used as termination. Derived criteria are affected by the kind of the problems and the used operators. There is no guarantee for accomplishing this criteria. An optimisation with a derived termination criterion should additionally have a direct criterion, to guarantee a termination [Pohlheim 2000].

Genetic algorithms differ from conventional optimisation methods (e.g. deterministic optimisation methods) by the following facts [Sivanandam and Deepa 2008]:

- GAs operate with coded versions of the problem parameters rather than the parameters themselves. Due to this fact different kinds of parameters can be used, e.g. strings, continuous and discrete values, or tree representations.
- GAs use a fitness function for evaluation rather than derivatives, which supports the application to any kind of continuous or discrete optimisation problem.
- GAs always operate on the whole population of solutions rather than just on a single solution. This improves the chance of reaching the global optimum and it also helps avoiding a local stationary point.
- GAs apply probabilistic operations instead of deterministic operations, which also supports to find the global optimum.

Due to these facts GAs are very suitable for supporting conceptional design, because especially in this phase different kinds of parameters are involved and the objective function is not continuous which prevents the application of deterministic optimisation methods.

One implementation of the ADT in the product development process is the Natural Optimisation Algorithm (NOA) [Jordan and Clement 2004]. NOA is a framework that contains several GA objects, which can be chosen and configured by the user. It was applied and verified in various projects with different partners to support the embodiment design of products at different levels of detail ([Mack et al. 1999], [Vajna et al. 2006], [Kittel et al. 2011a]).

In order to reflect the various requirements and conditions to be fulfilled by the emerging product, the ADT solution space contains taboo zones that can't be entered by possible solutions. The accompanying product model is described by using definable parameters of different views on the product [Kittel et al. 2010, 2011b]. Furthermore, the simultaneous optimisation approach was introduced to increase the efficiency of multi criteria evaluation chains at different levels of size and complexity [Wünsch et al. 2015].

The results of this research have one in common: they all require a product model. But especially in the early phases of conceptional design a product model doesn't exist. To close this gap this paper extends the application of the ADT by an approach for the flexible use in conceptual design by combining a morphological box and the optimisation framework NOA.

3. The Morphix framework

To resolve the weakness of the morphological box of possibly missing good solutions due to its size and complexity as well as to generate and evaluate concepts without preferred tendencies of the user, a computer-aided approach was made. Here, the method of the morphological box was combined with NOA to ensure the determination of the whole solution space in the morphological box.

To combine these two methods we developed the Morphix framework [Morphix 2016] as an open source fully flexible standalone tool using the programming language Python and the PySide library [Python 2015]. It can be used on any Windows-based system. The user interacts with the framework via mouse and keyboard, in which the keyboard is required only for the settings. Furthermore, it's possible to use it on a Windows tablet.

3.1 Usage scenario and front end

The main idea of Morphix is to gain the benefits of the morphological box and concurrently decrease its weaknesses. Figure 3 shows the basic usage scenario of Morphix.

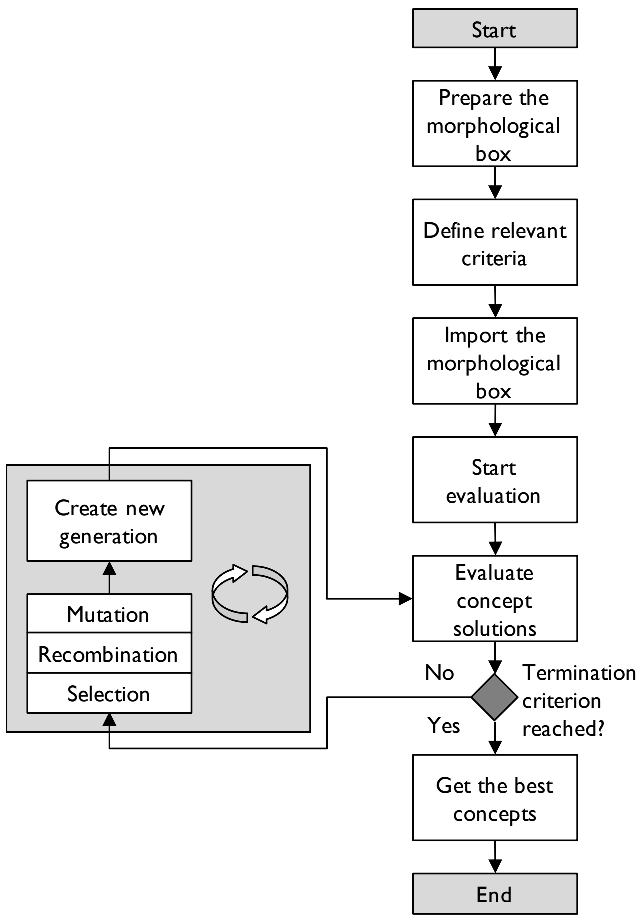

Figure 3. Usage scenario of Morphix

The first step is the creation of necessary input data, the morphological box is required in digital form. Therefore, the user has just to create folders in the Windows explorer named with the sub functions. Each folder contains image files to represent the solutions of the certain sub function. Furthermore, the user can specify the optimisation settings of NOA, e.g. mutation rate and population size.

The user needs to select relevant criteria for the evaluation process by giving criteria and their weight (e.g. performance). On the one hand predefined evaluation attributes based on the product attributes in

Integrated Design Engineering can be selected (cf. 3.3). On the other hand, it is possible to create custom criteria. Of course, a combination of both is possible.
The morphological box is created from the directory structure of the input data automatically and the evaluation of the concepts will start. A new window contains the morphological box on the left side and the evaluation criteria on the right side (Figure 4).

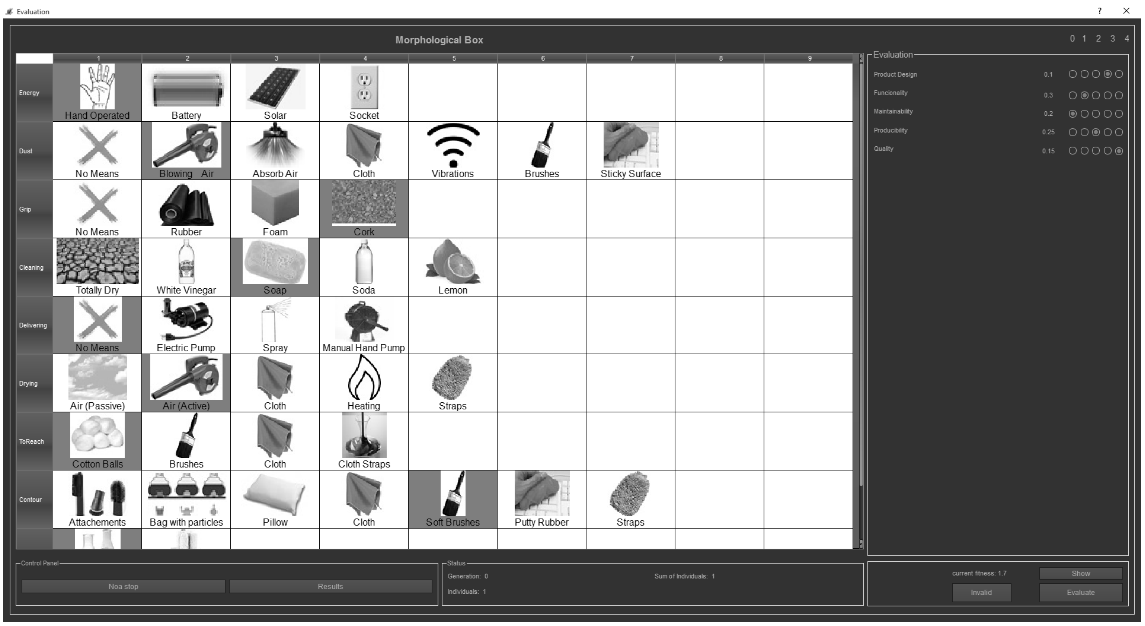

Figure 4. Morphix GUI during the evaluation process

To support the development of the GUI we used a questionnaire that was responded by students who have been dealing with morphological boxes in different projects. We found out that e.g. most of the students prefer a highlighted background in the morphological box to visualise a concept instead of using frames, only the images of the current concept or a list of the current features [Pilz 2015].
During the evaluation process the current concept is highlighted in the morphological box. The user can evaluate the current concept by rating it on a point scale using the values 0-4 on the predefined criteria [Feldhusen and Grote 2013]. We recommend doing the evaluation of the concepts in an interdisciplinary team to get different perspectives and expertise on the different concepts. If a concept contains an unrealisable combination of the sub solution, the user can mark this concept as invalid. After a few iterations (the number depends on the kind of the morphological box and its size) the GA recognizes the invalid solutions and combinations because they get a fitness of zero and the probability that an invalid solution occurs will decrease.
While the evaluation is running it's always possible to take a look on the current top three concepts. The evaluation goes on until one of the termination criteria is reached. After termination, Morphix shows the best concepts and all results sorted by fitness value.
The idea of combining a morphological box with an genetic algorithm is to generate and evaluate concepts without preferred tendencies of the user. Based on the rating the algorithm converges more and more towards the fitter sub solutions. Since we use a steady state based GA, the algorithm generates more fit concepts during the optimisation. The objective is, to gain the best concepts and to look simultaneously at random through the solutions-space. Thus the chance of getting only local optima is minimised.

3.2 Data interfaces and back end

To realise the process explained in section 3.1, the sub functions of the morphological box need to be transferred as discrete values to NOA, so NOA can interpret them. This data is processed continuously by interaction between Morphix and NOA. This interface is managed by Morphix, which can be seen as a front end to NOA. The primary data interfaces are shown in Figure 5.

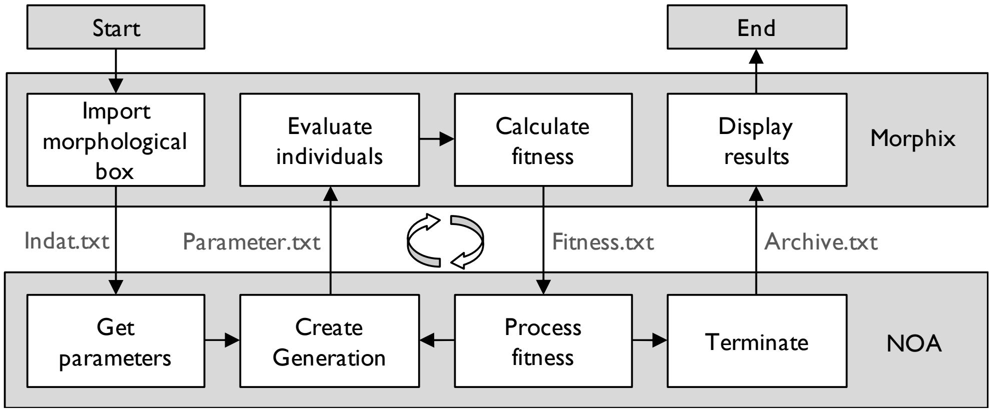

Figure 5. Interface between Morphix and NOA

At first Morphix starts importing the data to build up the morphological box from the directory that was set by the user. An optimisation parameter is created from each sub function (e.g. energy, cf. Figure 4). The solutions of the sub functions are represented by the parameter values. Dependent on the number of parameters, NOA creates the initial population and sends it to Morphix. Morphix reads the parameters and displays them in the morphological box for the user to evaluate the concept. After evaluating the concept using the predefined criteria, Morphix calculates the fitness by summating the ratings and their weighting using a weighted sum method:

$$Gw_i = \sum_{i=1;j=1}^{n;k} w_j \cdot m_{ij} \tag{1}$$

In the equation n represents the number of individuals, k the number of criteria, i the current individual/concept, j the criterion, w the weighting factor and m the evaluation value [Feldhusen and Grote 2013]. The fitness value is given back to NOA. If the termination criteria weren't reached, the next generation is established and transferred to Morphix.

For termination we use two criteria: convergence and number of generations. Convergence controls the improvement of the fittest individual. If the fitness value couldn't be increased over a number of generations (here 5) then the optimisation stops. To limit the total number of evaluations NOA stops after a certain number of generations (here 15). If one of the criteria has been reached, NOA finishes and writes all of the results to the archive file. Morphix can be used to display and analyse results.

3.3 Evaluation attributes

To support the evaluation process Morphix contains a set of predefined evaluation attributes based on the product attributes in Integrated Design Engineering ([Vajna et al. 2014], [Vajna 2015]). The basic attributes are divided into sub attributes which can be chosen by the user depending on the current product that will be evaluated. Table 1 shows the six attributes and their sub attributes.

Table 1. Product attributes in integrated design engineering

Product Design	Functionality	Handleability	Producibility, Availability	Maintainability	Sustainability
Harmony	Fulfilment of expectations of functionality	Expectation conformity	Manufacturing effort for in-house production	Inspectability	Sustainable materials

Pithiness & Originality	Robustness of functionality	Physiological ergonomics	Availability of purchased parts	Serviceability	Energy saving
Meaningfulness of product semantics	Efficiency of functionality	Efficiency in use	Product deployment	Overhaul ability	Dynamic value
Affordance		Functionality	Terms of availability	Improvability	Cyclability
					Social Sustainability

4. Case studies

During the development of Morphix, we used different examples of morphological boxes from literature and current student projects within the master's programme Integrated Design Engineering at the Otto-von-Guericke University Magdeburg, in which the users also provide feedback on the performance and usability.

Besides a cleaning device for delicate objects [Farrugia et al. 2014] and a vehicle parking sensor for cars (which is currently in progress), Morphix was applied to verify the benefits of its approach at the conceptual design of a 3D Printer for mass production. The morphological box was created to gain different high rated concepts. The project members just had to evaluate the concepts that were suggested by Morphix. The morphological box is shown in Figure 6.

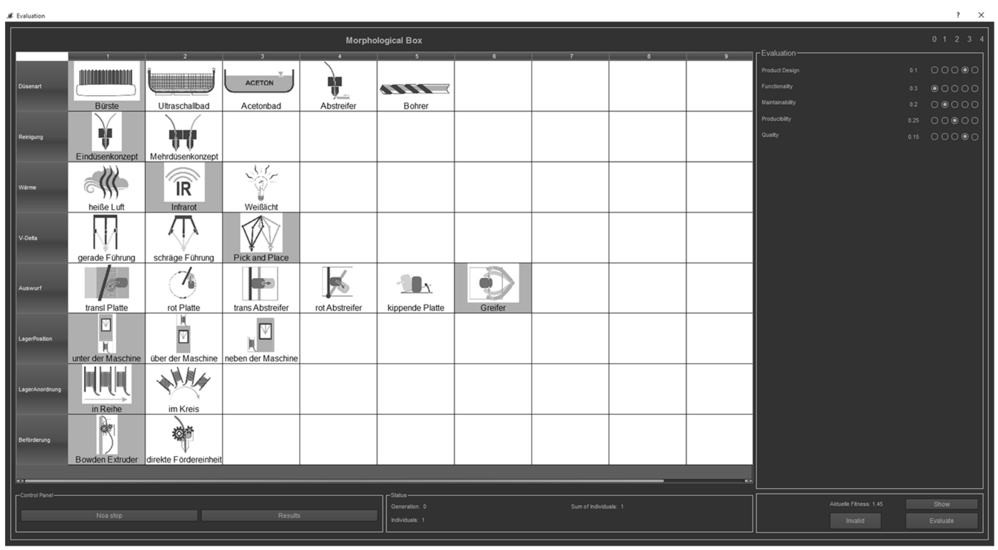

Figure 6. Morphix GUI during the evaluation process of different concepts of the 3D printer

Figure 7 shows the progression of the fitness value. The left diagram shows the progression of the maximum, minimum and average fitness value of a generation. The right diagram shows the progression of the fitness value sorted by rank in the certain generation.

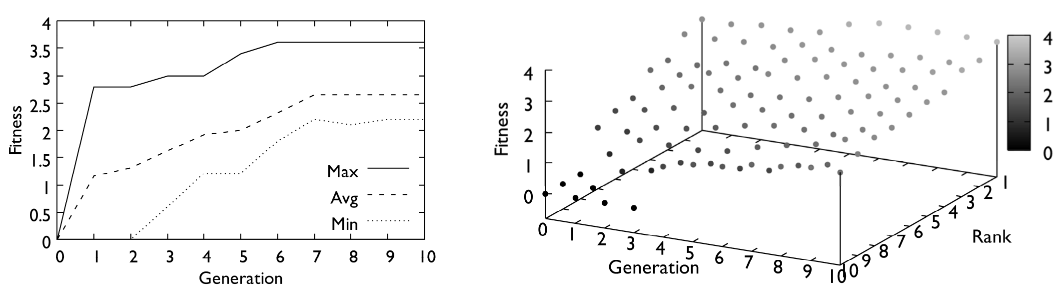

Figure 7. Progression of fitness value over generation and rank

As shown in Figure 7, the team needed 10 generations to get convergence (the fitness values couldn't be increased over the five generations from generation 6 to 10). Additionally, a population size of 10 was used which can be seen in the right diagram where a rank from 1 to 10 is shown. The chosen population size and number of generations to convergence represent a good compromise between ensuring to determine the whole solution space and reducing the number of total concept evaluations.
The diagrams show the continuous increase of the fitness value during the concept generation. The GA recognises good sub solutions (selection) and combines theses (recombination), while continuously ensuring diversity in a generation (mutation).
It can be concluded that the concepts, which are provided by NOA, converge with each generation in direction of the user's requirements, ideas, and wishes. The same tendency displays the minimum graph in the left diagram. Even until the ninth generation, the lowest fitness of each generation shows a continuous increase. Only the best fitness values catch their highest peak at the sixth generation. Until the algorithm stopped, any individual wasn't rated higher. Figure 8 shows the digital prototype of the final solution.

Figure 8. Digital prototype of the final solution

The complete evaluation process took ca. three hours. It was done in two meeting sessions of 1,5 hours each. Rating the first individuals took much more time than subsequent ratings because the team members had to get some routine in the rating process especially on discussing advantages and disadvantages of the certain concepts while considering each evaluation criterion.
After the sessions the opinions of the users were requested regarding the application and the support of Morphix for their work. It was proved, that the users have to go through the full solution space and unforeseen products can be generated.
On the other hand, the processing time was evaluated negatively. For the user it is not recognisable when the algorithm reaches convergence, because at any time an unforeseen highly rated solution can be generated and counting generations to convergence starts again from zero. The required time of the evaluation process can be very variable. Besides it was mentioned, that an absolute evaluation can be difficult [Pilz 2015]. To solve this problem the wish for a possible binary comparison to the previous concept was emerged. It would be advantageous even when not compatible sub solutions automatically are not generated.

5. Discussion

As shown in the case studies we developed a "self-learning" system by combining the well approved morphological box with a GA. The product concepts, which are provided by NOA, converge with each generation and a continuous increase of the fitness value can be ensured. Contemporaneous diversity in a generation can be guaranteed and the whole solution space is considered.
The main disadvantage of a GA is the large number of evaluations. The complete evaluation process took about three hours and the team rated 144 different concepts, including 30 invalid concepts. The difference between the 144 rated concepts and the 100 individuals that are shown in Figure 7 results from post generation operations, e.g. vary best, offer best, and reinitiate worst.
Although the number of evaluations is high, it is still low compared to the number of all possible solutions which is 6480. All these possible solutions can't be evaluated by the designer manually. Using Morphix we evaluated only 2.2 % of all possible solutions and reached convergence which yields to

several high rated concepts. Furthermore we appreciate the extensive evaluation process as a chance for thinking "out of the box", because the discussions in the evaluation sessions yield to some new insights on the product.
Extending the morphological box in case of developing a product that is more complex will increase the number of all possible solutions as well as the number of evaluations needed to reach convergence. This enlargement will limit the application of a morphological box in general and the application of Morphix, too. In this case it could be suitable to use both incrementally on different levels of complexity or in different domains.
Another challenge for the team members is to stay impartial and objective during the evaluation process. To ensure this it is recommended to do the evaluation by an interdisciplinary team where the different concepts can be discussed. Since GAs are able to deal with binary comparisons, it should also be possible to evaluate the concepts by comparison, which can be more intuitive than the objective rating. Binary comparisons haven't been tested yet.
Invalid solutions have to be marked by the user and rated with a fitness of zero. Another possibility would be to detect these invalid solutions automatically after a new generation was created and rate them automatically with zero, e.g. by underlying rules and conditions from solution catalogues e.g. created by Roth [1996, 2000, 2001] or predefined ratings and combinations [Katzwinkel et al. 2015]. We decided to disregard this in order to avoid restricting the user's creativity and getting new insights, because conceptual design profits from intuition, creativity and thinking "out of the box". Due to this fact, we only use simple images to represent the sub solutions and the "pure" morphological box, so that Morphix is more flexible and can be used for different products at different levels of abstraction.
On the other hand semi or full automatic evaluations could be a great chance to simplify the evaluation process for the user, especially when evaluating similar products over and over again. The sub solutions could be stored in a database after the user created them. Here we see analogies to AskNature [AskNature 2015] and similar databases. Furthermore, the sub solutions could include meta data from previous evaluations, e.g. successful ratings and combinations. This meta data could be used for semi or full automatic evaluation of the concepts.

6. Conclusion and outlook

In this paper we introduced the Morphix framework to support conceptual design in an evolutionary way by using aspects of the Autogenetic Design Theory. Morphix combines the morphological box that is a commonly used method to generate concepts and the optimization framework NOA. Morphix generates concepts and the designer rates the proposed concepts by predefined or custom evaluation criteria. It was applied in different use cases from literature and current student projects within the master's programme Integrated Design Engineering (IDE) at the Otto-von-Guericke University Magdeburg. We found that the application of Morphix supports thinking "out of the box" and getting new insights on the product since the genetic algorithm regards the whole solution space.
Morphix is distributed as an open source tool to gain broad feedback by different users [Morphix 2016]. Since one major challenge for the user is to stay impartial and objective, binary comparisons could be suitable to simplify the concept evaluation.
Genetic algorithms are able to deal with binary comparisons. Further research could be done on decentralising the preparation of the morphological box and the evaluation process by implementing e-collaboration tools. This could be a possibility to reduce the effort to get the relevant people at one table for the evaluation session and to integrate more people to the evaluation.

References

AskNature online, Available at <http://www.asknature.org/>, 2015, [Accessed 02.12.2015].

Bercsey, T., Vajna, S., "Ein Autogenetischer Ansatz für die Konstruktionstheorie. Beitrag zur vollständigen Beschreibung des Konstruktionsprozesses", CADCAM Report, Vol.13, No.2, pp. 66-71 and Vol.14, No.3, 1994, pp. 98-105.

Farrugia, L., Roa Castro, L., Ben Beldi, N., Wünsch, A., Hagman, J., Drágár, Z., "Cleaning Genie: An International Case Study in Integrated Product Development", Proceedings of 10th International Workshop on Integrated Design Engineering, Magdeburg, Germany, 10.-12.09, 2014.

Feldhusen, J., Grote, K.-H., "Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung", 8. vollst. Überarb, Springer-Verlag Berlin Heidelberg, 2013.

Gerdes, I., Klawonn, F., Kruse, R., "Evolutionäre Algorithmen", Friedrich Vieweg & Sohn Verlag Wiesbaden, 2004, pp. 26-80.

Großklaus, R., "Von der Produktidee zum Markterfolg", Springer Gabler Wiesbaden, 2014, pp. 112-115.

Jordan, A., Clement, S., "Handbuch für das Konstruktionssystem NOA", 2004.

Katzwinkel, T., Heller, J. E., Schmid, A., Schmidt, W., Löwer, M., Feldhusen, J., "Improving Design Methodology: Systematic Evaluation of Principle Synthesis", Proceedings of ICED15, Vol. 2, Milan, Italy, 27.-30.07, 2015.

Kittel, K., Bercsey, T., Vajna, S., "Designing the Solution Space for the Autogenetic Design", Proceedings of DESIGN 2010, Marjanovic, D. (Ed.), Zagreb, 2010.

Kittel, K., Fritzsche, M., Vajna, S., "The Autogenetic Design Theory and its practical application", Int. Journal of Design Engineering, Vol.4, 2011a, pp. 197 – 213.

Kittel, K., Hehenberger, P., Vajna, S., Zeman, K., "Product Model of the Autogenetic Design Theory", Proceedings Vol. DS68-4, ICED11, Kopenhagen, 2011b, pp. 309-318.

Mack, P., Wegner, B., Bercsey, T., Vajna, S., Kickermann, H., Kellner, P., "Computer Aided Optimization of Cylinder Head Inlet Ports Using Genetic Algorithms", Lindemann, U., Birkhofer, H., Meerkamm, H., Vajna, S. (Eds.), Proceedings of ICED 99, München, 1999.

Morphix online, Available at <https://sourceforge.net/projects/morphixx/>, 2016, [Accessed 14.03.2016].

Pilz, F., "Unterstützung der Konzeptentwicklung durch den Einsatz von genetischen Algorithmen", Otto-von-Guericke University Magdeburg, Bachelor Thesis, 2015.

Pohlheim, H., "Evolutionäre Algorithmen - Verfahren, Operatoren und hinweise für die Praxis", Springer-Verlag Berlin Heidelberg, 2000, pp. 26-65.

Python online, Available at <https://www.python.org/>, 2015, [Accessed 30.11.2015].

Roth, K., "Konstruieren mit Konstruktionskatalogen", 3. Aufl. Bd III. Verbindungen und Verschlüsse, Lösungsfindung, Springer Berlin, 1996.

Roth, K., "Konstruieren mit Konstruktionskatalogen", 3. Aufl., Bd I. Konstruktionslehre, Springer Berlin, 2000.

Roth, K., "Konstruieren mit Konstruktionskatalogen", 3. Aufl., Bd II. Konstruktionskataloge, Springer Berlin, 2001.

Sivanandam, S. N., Deepa, S. N., "Introduction to Genetic Algorithms", Springer Berlin Heidelberg, 2008.

Vajna, S., "Attributes in Integrated Design Engineering - A new Way to Describe Both Performance Capability and Behaviour of a Product", Proceedings of ICED15, Vol. 2, Milan, Italy, 27.-30.07, 2015.

Vajna, S., "Integrated Design Engineering: Ein interdisziplinäres Modell für die ganzheitliche Produktentwicklung", Springer Berlin Heidelberg, 2014.

Vajna, S., Clement, S., Jordan, A., Bercsey, T., "The Autogenetic Design Theory: An Evolutionary View of the Design Process", Journal of Engineering Design, Vol.16, No.4, 2005, pp. 423-440.

Vajna, S., Edelmann-Nusser, J., Kittel, K., Jordan, A., "Optimisation of a bow riser using the Autogenetic Design Theory", Tools and Methods of Competitive Engineering, Ljubljana/Slovenia, 2006, pp. 593-602.

Wünsch, A., Jordan, A., Vajna, S., "Simultaneous Optimisation: Strategies for using Parallelization Efficiently", Proceedings of ICED15, Vol. 6, Design Methods and Tools – Part 2, Milan, Italy, 27.-30.07,2015.

Zwicky, F., "Entdecken, Erfinden, Forschen im Morphologischen Weltbild", Droemer- Knaur München 1966.

Andreas Wünsch, Dipl.-Ing.
Otto-von-Guericke University Magdeburg, Information Technologies in Mechanical Engineering
Universitätsplatz 2, 39106 Magdeburg, Germany
Email: andreas.wuensch@ovgu.de