

# Morpheus: The Development and Evaluation of a Software Tool for Morphological Matrices

Julian Martinsson Bonde<sup>1</sup>, Adam Mallalieu<sup>1</sup>, Massimo Panarotto<sup>1</sup>, Ola Isaksson<sup>1</sup>, Lars Almfelt<sup>1</sup>, Johan Malmqvist<sup>1</sup>

<sup>1</sup>*Chalmers University of Technology*  
*Julianm@chalmers.se*

## Abstract

The Morphological Matrix is a well-known and established method for synthesising concepts in engineering design. With the aim to improve its applicability, the method has been implemented into a software tool, which then has been introduced in engineering courses at Chalmers University of Technology. In this paper, the tool and its most important functions are presented and discussed, along with experiences from its use in design projects. It was demonstrated that the tool can ease the process of creating and modifying morphological matrices. Furthermore, the tool demonstrated the ability to generate and keep track of large quantities of solutions, which has previously been difficult as morphological matrices are typically hand drawn or created using an Excel sheet. However, less desired effects were encountered as the tool was perceived by some students as more of a screening and down-selection method, rather than as a tool for synthesising concepts. The learnings and the way forward in education and the industrial context are discussed.

**Keywords:** *Concept synthesis, design support, morphological matrix*

## 1. Introduction

The conceptual design phase is a critical phase in engineering design due to its high impact on product development cost (Ullman, 1992). It is typically characterised by multiple quick design iterations, and as the concepts are developed, new information surfaces which prompts new iterations (Ulrich & Eppinger, 2020). This emphasises the ability to explore many different concepts as quick as possible, such that needed information is collected as soon as possible. The Morphological Matrix (MM) (Pahl et al., 2007) is a well-established method originally proposed by Zwicky (1967) that supports systematic formation of concepts (Weber & Condoor, 1998). The MM is versatile and have been adapted to be used in several different ways in engineering design (Diaz Tena et al., 2021; Fargnoli et al., 2006; Ölvander et al., 2009). The MM has also been proven to be useful in practice, both to generate synthesised concepts of subsolutions, and to stimulate a synergy analysis (Almfelt, 2005a).

## Synthesis procedure

(X) Synergy analysis X



Solution principles / Function/ Sub-system	1	2	3	4	5	6
1 Transfer Side Impact Forces	S1.1 Separate block 	S1.2 Integrated in trim 	S1.3 Multiple storage 	S1.4 Box a la Audi 	S1.5 SIPS-speaker 	S1.6 Box & SIPS in seat 
2 Audio (principles, locations, geometry)	S2.1 Current local 	S2.2 Bass in side wall 	S2.3 Bass below seat 	S2.4 In parcel shelf 	S2.5 Bass in backrest 	S2.6 Upper front area 
3 Daily Life Storage	S3.1 Conventional 	S3.2 Foldable 	S3.3 Net (vinyl) pocket 	S3.4 X-drawer 	S3.5 Y-drawer 	S3.6 Z-drawer 
4 Open/Close Window	S4.1 Cross arm 	S4.2 Double rail 	S4.3 Wire in frame 	S4.4 Single wire rear 	S4.5 S&A 96 pivot 	S4.6 Lamella slide window 
5 Provide for Ingress/Egress	S5.1 Separate inner release 	S5.2 Inner release in armrest/grab handle I 	S5.3 Inner release in armrest/grab handle II 	S5.4 Inner release in grab handle I 	S5.5 Inner release in grab handle II 	S5.6 Touch release 
6 Structural/Architectural Principles	S6.1 Sheet metal 	S6.2 Frame-work in Mg 	S6.3 Super-integration 	S6.4 Plastic box 	S6.5 Function module 	S6.6 Flat-wire harness 
7 Provide for Ergonomic & Easy Assembly	S7.1 Clips 	S7.2 Hooks 	S7.3 Central fixing 	S7.4 Slide-in 	S7.5 Dry-side wire harness connection 	S7.6 Light colours 

Figure 1 Morphological Matrix applied in the automotive industry (Almfelt, 2005a). Included with permission from the author.

In engineering design, the creation of an MM is commonly preceded by a functional decomposition. To perform a functional decomposition the primary function of the potential product is divided into subfunctions (SFs) that needs to be performed to fulfil the primary function (Ulrich & Eppinger, 2020). The engineers then generate subsolutions (SSs) for each SF, during a creative session such as a brainstorming activity. The MM is then configured by listing all SFs within one matrix column. Each row is then populated with the SSs of the corresponding SF. A solution to the primary function (also referred to as a *concept*) can thus be synthesised by combining one SS for each SF. This is exemplified in Figure 1, which depicts one section of an MM created by a Swedish automotive company to explore car door variants. Especially in early phases of the design, insights gained through design activities often lead to updates and refinements of both the SF's and the SS's. Smooth creation and updating of an MM are consequently desired.

The MM has traditionally been implemented and used by hand, or using tools such as Excel and PowerPoint. Implementing MMs can thus be a tedious and time-consuming process. Two of the most prominent issues are: 1) The MM often needs to be changed over time. 2) The user needs to use the MM to keep track of many solutions. The MM often needs to change because, for instance, a new subfunction might emerge, or new subsolutions discovered. Changing an MM that has been drawn by hand, or using an Excel sheet, can be problematic and require an unnecessary amount of work. Consequently, it has been observed that students in engineering design sometimes avoid adopting changes into their MMs, even if the initial version was found lacking. It has also often been found that the students refrain from generating more than a few concepts, since keeping track of multiple solutions in an MM can be difficult. Thus, large portions of the design space remain unexplored. These issues hamper the design process, and the identified SF's and SS's risk becoming obsolete.

There is currently a transformation in industry, where the need for digital design tools is increasing (Hallstedt et al., 2020; Isaksson et al., 2020) along with the need for strengthening the skills of future engineers in using such tools. One of the overall goals of engineering design research is to develop methods and tools that support industry in developing successful products (Blessing & Chakrabarti, 2009). However, introducing such methods and tools in industry has

been observed to be problematic (Araujo et al., 1996; Eder, 1998). It has also previously been observed that computerised design methods and tools can entail several implications (Mallalieu et al., 2022). Altogether, this makes it justified to develop a software tool for MM while investigating potential benefits and downsides. This entails the following research questions:

*RQ1: How can the Morphological Matrix method be developed into a software tool?*

*RQ2: What are the potential downsides and benefits of the Morphological Matrix method as a software tool?*

To address RQ1, a software called *Morpheus* (Martinsson Bonde, 2021) has been developed. The tool has been practically implemented and used in university courses in product development for two years. The process of developing Morpheus is presented in this paper. Furthermore, to address RQ2 a discussion of the implications of introducing this software as an alternative to a historically non-computerised process is presented. The remainder of this paper is structured as follows: First, the research method used is presented. Second, the developed MM software Morpheus is presented. Third, an evaluation of the tool is presented. Fourth, the potential benefits and limitations of the tool are discussed. Finally, conclusion and future recommendations are presented.

## **2. Research Method**

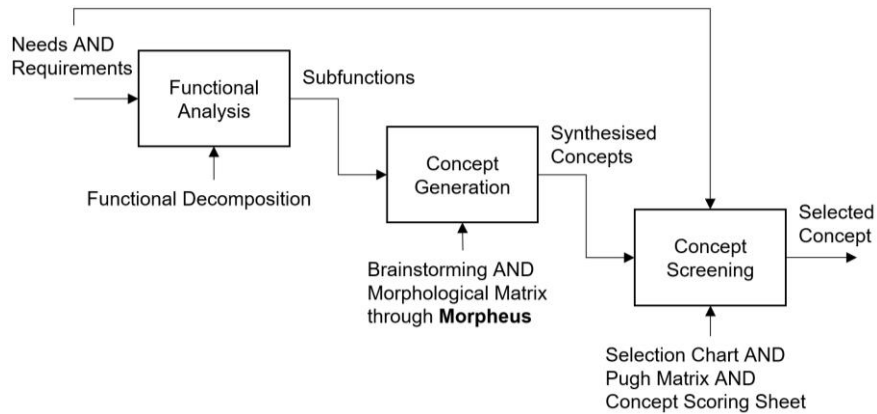
In this section the research methods will be presented. Section 2.1 contains a short description of how Morpheus was developed. Section 2.2 describes how the tool was evaluated. In total, an estimate of 500 students have been introduced to the tool in nearly 100 engineering projects. After two years of project use, an analysis of the findings was conducted, followed by a discussion of the further exploitation. These results are reported in the remainder of the paper.

### **2.1 Morpheus software development**

Following several years of experience from teaching and supervising student design projects at Chalmers University of Technology, the need to develop a software tool was compiled into an implementation specification. The basic method as published by (Pahl et al., 2007) was used to define the method, and the associated requirements were based on the needs expressed by teachers and students in a product development course. The first version of the tool was released in 2020. It was written using JavaScript on top of the Electron framework (OpenJS, n.d.). The developed tool was evaluated using surveys involving students of the first years, and through a focus group with course teachers. The feedback gained through the evaluations resulted in updates to the software to add new functions, refine existing functions, and to correct erroneous software behaviour.

### **2.2 Evaluation through university design course**

Two courses in product development for first year students studying mechanical engineering were selected for testing Morpheus. In these courses, the students work in groups of four to six, to learn about the process of product development through practical application. The groups started out by formulating a problem, and then go through a basic product development process largely based on Ulrich et al., 2020, with the end goal of creating a prototype and a description of the final product. In these courses the students utilise several design methods and MMs are specifically used for concept generation. In 2020 and 2021, the students were instead asked to use Morpheus for creating their MMs. The overall process implemented by the students is visualised in Figure 2, where Morpheus is used in the concept generation activity.



**Figure 2. Morpheus positioned in relation to the product development process used in the university courses.**

### 2.2.1 Questionnaire with mechanical engineering students

After the course had concluded in 2020, one student per group were asked to answer a questionnaire. The primary purpose of the questionnaire was to evaluate how the students perceived Morpheus, and if the software performed as intended. Some of the key questions from the questionnaire can be seen in Table 1. In total, 12 of 36 groups answered the questionnaire.

**Table 1: Key questions included in the questionnaire that was sent out to students in the 2020 course.**

Question	Possible answers
Which operative system did you install Morpheus on?	"Windows", "Mac OS", "Linux"
Did your group use an image exported from Morpheus in your final report?	"Yes", "No", "We have not yet decided"
Did you encounter any "bugs" in Morpheus?	"Yes, 10 or more", "Yes, 4-9", "Yes, 1-3", "No, we did not encounter any bugs"
How many subfunctions did you include in your morphological matrix?	"1-3", "4-6", "7-9", "More than 9"
How many pictures did you attach to your morphological matrix?	"1-3", "4-6", "7-9", "More than 10", "We did not attach any pictures"
How many solutions did you generate?	"0", "1-10", "11-30", "31-50", "51-70", "71-90", "91-100", "100<"
"Morpheus is easy to learn"?	Likert scale, 1-5 from "I fully agree" to "I fully disagree"
"Using Morpheus resulted in the group identifying solutions we otherwise wouldn't have thought of"	Likert scale, 1-5 from "I fully agree" to "I fully disagree"
What can be improved in Morpheus next year?	Free text

### 2.2.2 Focus group with design course supervisors

A focus group was conducted to evaluate Morpheus' usability further. The focus group involved seven of the eight teachers in the course who had supervised the projects and observed how Morpheus had been used. These teachers included three PhD students, one senior lecturer, and three professors. The purpose of the focus group was to evaluate the tool based on typical criteria a design method and tool should strive to fulfil (Almefelt, 2005b). Those criteria include that the tool should be easy to understand, easy to use and learn; and that it should have a positive and preferably measurable effect on the outcome.

## 3 Results

In this section the results from the tool development and evaluation will be presented.

### 3.1 The morphological matrix creation software - Morpheus

Morpheus is a software that allows the user to create and modify morphological matrices through a graphical interface. In this section, the software and its capabilities are described.

#### 3.1.1 Core functionalities

Morpheus was designed to make it as easy as possible to create and edit MMs. When launching the software, the user is greeted by an empty Matrix and a toolbar. The user can then easily add subfunctions and subsolutions by clicking on the "+ SF" and "+ SS" buttons in the matrix (see Figure 3). The user can write text in the cells by selecting them. Hovering with the cursor over a cell displays a context menu which allows the user to add images to SSs, delete the SS/SF, and move the matrix row up or down.



Figure 3 The user can add new SSs or SFs to the matrix by clicking on the "+SS" or "+SF" buttons.

Once the matrix has been populated with SSs it is possible for the user to mark pairs of SSs as incompatible. For instance, suppose the user is creating an MM to investigate different ways of creating a land vehicle, and thus has one SF called "Store energy", and another called "Convert stored energy to motion". An SS to "Store energy" can be "Fuel tank", and an SS to "Convert stored energy to motion" can be "Electric motor". However, an electric motor has no need for fuel. Consequently, the user is not interested in any solutions where the *fuel tank* has been combined with the *electric motor*. Therefore, a feature has been implemented allowing the user to mark these two SSs as incompatible, ensuring that they can't be combined. Once an SS is selected, all SSs that are incompatible are disabled, which is visualised using a yellow "X"-symbol on top of the cell. Each time a new incompatibility is added, or a new SS implemented, a counter in the bottom left corner (Figure 4) gives the user an indication of how many unique solutions can be generated from the matrix.

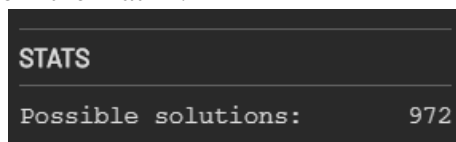
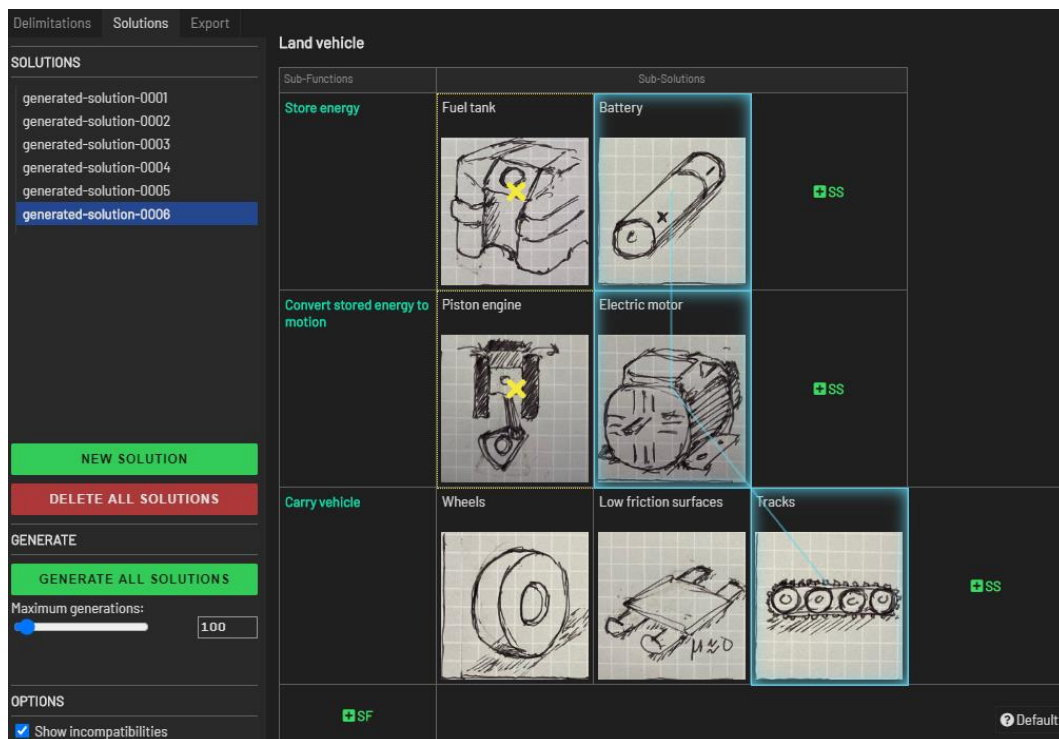


Figure 4 The software keeps track of the number of possible solutions with regards to incompatibilities.

When all incompatibilities are in place, the user can elect to either generate all possible solutions using the "Generate solutions"-button, or manually create solutions. Figure 5 shows a screenshot of a matrix in which six solutions has been automatically generated. Once the user is satisfied with the results, the matrix or its solutions can be exported either as a CSV-file or as an image. The next step is typically to go through the generated solutions one-by-one and create sketches and descriptions of each concept. These activities are performed outside of the software and allows the team to interpret what each subsolution combination actually means. On the other hand, if the user is unable to finalize the matrix within a single session it is also

possible to save the work-in-progress matrix in a “.morph”-file, allowing the user to resume work on their project again at another time.

This tool not only makes it easier to make changes to the matrix compared to if done by hand, but it also assists the user in keeping track of created concepts (solutions). Each created solution is stored in a list on the left-hand side of the screen. Selecting a solution from this list highlights it in the matrix. Furthermore, by allowing users to mark SSs as incompatible, the process of creating solutions is simplified as the software will not allow the user to create unfeasible solutions by mistake. This can be especially useful in matrices with a high number of SFs where it can be difficult to keep track of incompatibilities manually.

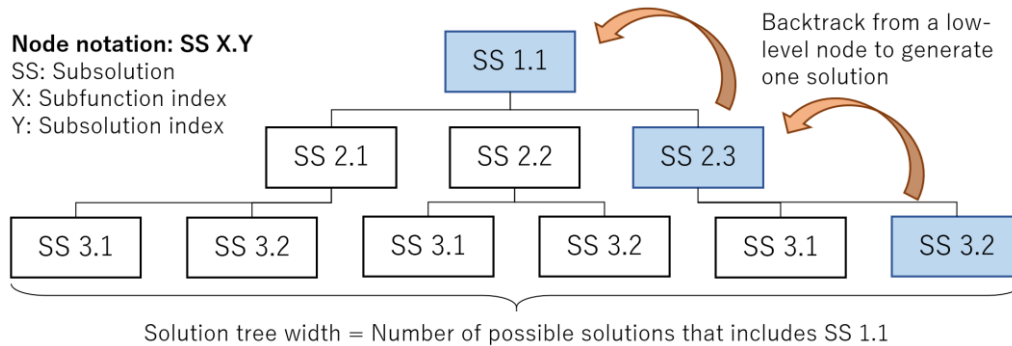


**Figure 5. Example of Morpheus in use. The blue cells represent the subsolutions of the selected solution. The cells with yellow crosses are incompatible with the selected subsolutions.**

### 3.1.2 Automated solution generation and counter

As previously described, when the user adds and removes cells, or introduces new constraints, a counter on the screen gives the user immediate feedback of how many solutions the matrix contains. The number of possible solutions in an MM without any constraints can easily be calculated by multiplying the number of solutions for each SF. However, when introducing incompatibilities between SSs, then analytical calculations are no longer feasible. Instead, the software creates a specialized data structure, referred to here as a *Solution Tree*. One solution tree is created for each SS for the top-level SF. The tree is built by starting from one of the top-level SSs, and then creating “branches” to each compatible SS in the following SF. This process is repeated for each identified compatible SS until the algorithm reaches the final SF. By only branching together SSs that are compatible with all previously encountered SSs, the tree avoids solutions that are unfeasible due to incompatibilities. The width of the lowest level of the tree equals the number of possible solutions that incorporates the initial SS. The tree building algorithm thus needs to be run once for every SS for the first SF to get the total solution count.

The same algorithm, with an extra step, is also used to generate all possible solutions. By ensuring that each node in the tree is aware of its parent node, then all solutions can be generated by backtracking from the low-level nodes. This automatic solution generation can be triggered by the user by clicking the “Generate all solutions”-button. The tree, the solution counting, and the solution generation algorithm are depicted in Figure 6.



**Figure 6 Solution tree data structure in principle. Each node in the tree is a subsolution. Each branch in the tree is a connection between two compatible subsolutions.**

### 3.2 Evaluation

A questionnaire along with a focus group was conducted to evaluate Morpheus, and the results from these are summarized in this section.

#### 3.2.1 Questionnaire

The questionnaire revealed that 9 out of 12 groups used a Windows machine to install Morpheus. The remaining 3 groups used Mac OS. The same ratio of students also used an exported image from Morpheus in their final project report. A majority of the groups (58.3%) did not encounter any bugs, while the remaining groups encountered a low number of bugs. Most of the identified bugs were of a superficial nature. However, one critical issue was identified which on rare occasions caused the save-function to fail, resulting in the students losing their work.

Regarding how the students used the software, the questionnaire showed that most groups created a matrix that contained around 4-6 subfunctions. Notably, only half of the groups used the built-in feature that allowed them to attach images to subsolutions. Furthermore, a majority of the groups generated more than 31 solutions.

The students had a generally positive attitude towards Morpheus when it was first introduced. All answers to the questionnaire stated that Morpheus was easy to learn. At the same time, the opinion on whether or not Morpheus had been of assistance in identifying new solutions was very divided, though a majority of the groups did not think so. Multiple suggestions for improvements were submitted. The most prominent suggestions were that the image export from Morpheus should be of a higher quality, and that more tutorials should be created showing how to use the software.

#### 3.2.2 Focus group

The focus group discussed how the implementation of Morpheus had affected the university course. There were some observations that several teachers had in common. For instance, there were some student groups that had been observed using the MM as a strict screening method, while also referring to MM as a screening method in the project reports. In addition to this, the students also used Morpheus to screen solutions against the formulated requirements lists



created earlier in the course. However, in the course, screening using the requirements list should rather be done using an evaluation chart (Pahl et al., 2007), a Pugh matrix, or concept-scoring/Kesselring matrix (Ulrich & Eppinger, 2020), after the concepts have been generated. It was also observed that some of the students referred to the software “Morpheus”, rather than the method “morphological matrix”, when describing their systematic concept generation process.

There were also students that had issues with seeing the connection between different activities in the product development process used in the course. The students are supposed to use the identified subfunctions from the functional analysis as input to Morpheus, as visualized in Figure 2. This was not always the case, as there were groups that used different subfunctions in Morpheus compared to what was identified in the functional analysis. Furthermore, the output of Morpheus is intended to be used as input to the concept screening. The output of an MM should typically be synthesised concepts consisting of several combined subsolutions, a description of the concept, and preferably a sketch. Instead, the output produced by the students using Morpheus was sometimes merely a combination of words that had been automatically generated by the software. A typical example of such a case is illustrated in Figure 7.

<b>generated-solution-0001</b>	<b>generated-solution-0002</b>	<b>generated-solution-0003</b>	
Electrically	Electrically	Electrically	
Chemically	Mechanically	Thermally	
Electrically	Electrically	Electrically	
<b>generated-solution-0006</b>	<b>generated-solution-0007</b>	<b>generated-solution-0008</b>	
Electrically	Electrically	Electrically	
Thermally	Chemically	Mechanically	
Mechanically	Thermally	Thermally	
Sub-Functions	Sub-Solutions		
Generate energy	Electrically	Mechanically	Thermally
Store energy	Chemically	Mechanically	Thermally
Transfer energy	Electrically	Mechanically	Thermally

**Figure 7. Example of student results after misusing the method, and the software.**

In the course material, multiple strategies are suggested on how to synthesise solutions using an MM. Rather than utilizing these strategies, it was in some cases observed that students used intuition to generate a set of solutions, or made use of the automatic solution generation function without properly defining the generated concepts. There was no reflection about what it meant to combine several sub-solutions, or why certain combinations work better or worse. Furthermore, there were few groups that made use of pictures in Morpheus, making it more difficult for the students to convey the meaning of their solutions.

It could however be seen that several groups made use of more iterations in the concept generation step involving Morpheus, since it was easy to make changes in the MM using the software. Morpheus was also generally perceived as positive and as a helpful tool by students, as it eased some of the manual work.

## 4 Discussion

In general, there were some observed problems while using Morpheus. However, these problems mainly concerned how it was implemented and used by students, and not the software



itself. It should also be noted that the students that made use of the tool were first year students within the mechanical engineering program, and that the design project is likely their first encounter with the morphological matrix method. This does however highlight the importance of understanding the method to obtain an appropriate result. The software Morpheus can be argued to be easy to understand, use, and learn. However, more guidance in how to implement an MM is needed, such that the students gain an understanding of the fundamental method *before* attempting to use the software.

The students also had issues in being able to see the connection between the steps before and after the concept generation step, where Morpheus is used. This led to some incompatibility issues between the activities. This led to the conclusion that there needs to be more functionalities in Morpheus that assists the user in integrating the tool with upstream and downstream activities. How this can be achieved in practice is further presented in Section 5.

Morpheus was both developed and evaluated as a tool used in conjunction with other methods, which can increase the ability of a design tool to be used in industry. Morpheus was however evaluated in a university context which is not deemed enough to validate its usability in industry (Gericke et al., 2020). Morpheus is also planned to be further developed based on the issues observed, and will also be further tested and evaluated as part of a toolbox in industry (Hallstedt et al., 2022).

## 5 Conclusion and outlook

In this paper, a software tool for creating morphological matrices was presented. An answer to **RQ1** - *“How can the Morphological Matrix method be developed into a software tool?”* - was thus proposed in the form of “Morpheus”. It was developed at Chalmers University, where it was also applied in two courses in product development. To answer **RQ2** - *“What are the potential downsides and benefits of the Morphological Matrix method as a software tool?”* - a thorough evaluation was conducted in the form of a questionnaire sent out to the students, and a focus group together with the teachers of one of the courses.

Based on the evaluation presented in this paper it's possible to see ways in which Morpheus can be improved: 1) The students need to understand that the functions in the morphological matrix are the same as the ones identified during the process of functional decomposition. One way of assisting the students in understanding this is to develop a functional modelling tool that is to be used before creating the matrix. The functions created in that view could be directly imported into the matrix, thus ensuring a one-to-one mapping of functions from the functional decomposition to the morphological matrix. 2) The students need to understand that the solutions extracted from the morphological matrix are concepts that needs to be further defined. Instead of exporting a CSV-table with all the subsolutions, one possible solution can be to ask the user to write a few sentences about each solution such that it is concretised beyond the simple combination of subsolutions. 3) The user needs to be encouraged to sketch subsolutions. However, such an encouragement needs to be done outside of the tool, and be enforced by the teachers in the courses. 4) Due to a recent update to Mac OS, the tool is no longer functional on that platform. However, the tool should be accessible to any user, no matter the platform they use. To enable this, Morpheus is to be converted to a web-based tool such that it works independently of operative system.

Morpheus is available as an open-source project at <https://github.com/johnmartins/morpheus>.

## References

- Almefelt, L. (2005a). Balancing properties while synthesising a product concept--a method highlighting synergies. *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia, 15.-18.08. 2005*.
- Almefelt, L. (2005b). *Requirements-driven product innovation: methods and tools reflecting industrial needs. PhD Thesis*. Chalmers tekniska högskola.
- Araujo, C. S., Benedetto-Neto, H., Campello, A. C., Segre, F. M., & Wright, I. C. (1996). The Utilization of Product Development Methods: A Survey of UK Industry. *Journal of Engineering Design*, 7(3), 265–277. <https://doi.org/10.1080/09544829608907940>
- Blessing, L. T. M., & Chakrabarti, A. (2009). DRM, a Design Research Methodology. In *DRM, a Design Research Methodology*. Springer London. <https://doi.org/10.1007/978-1-84882-587-1>
- Diaz Tena, A., Schögggl, J.-P., Reyes, T., & Baumgartner, R. (2021). EXPLORING SUSTAINABLE PRODUCT DEVELOPMENT PROCESSES FOR A CIRCULAR ECONOMY THROUGH MORPHOLOGICAL ANALYSIS. *Proceedings of the Design Society*, 1, 1491–1500. <https://doi.org/10.1017/pds.2021.410>
- Eder, W. E. (1998). Design Modeling-A Design Science Approach (and Why Does Industry Not Use It?). *Journal of Engineering Design*, 9(4), 355–371. <https://doi.org/10.1080/095448298261499>
- Fargnoli, M., Rovida, E., & Troisi, R. (2006). The morphological matrix: Tool for the development of innovative design solutions. *4th International Conference on Axiomatic Design, ICAD*, 1–7.
- Hallstedt, S. I., Isaksson, O., & Öhrwall Rönnbäck, A. (2020). The Need for New Product Development Capabilities from Digitalization, Sustainability, and Servitization Trends. *Sustainability*, 12(23). <https://doi.org/10.3390/su122310222>
- Hallstedt, S. I., Isaksson, O., Watz, M., Mallalieu, A., & Schulte, J. (2022). Forming Digital Sustainable Product Development Support. *Proceedings of the NordDesign 2022 Conference, NordDesign 2022*.
- Isaksson, O., Eckert, C., & others. (2020). Product Development 2040: Technologies are just as good as the designer's ability to integrate them. *Design Society Report DS107*. <https://doi.org/https://doi.org/10.35199/report.pd2040>
- Mallalieu, A., Hajali, T., Isaksson, O., & Panarotto, M. (2022). The Role of Digital Infrastructure for the Industrialisation of Design for Additive Manufacturing. *Proceedings of the Design Society*, 2, 1401–1410. <https://doi.org/10.1017/pds.2022.142>
- Martinsson Bonde, J. (2021). *Morpheus* (0.6.0). <https://doi.org/10.5281/ZENODO.6502828>
- Ölvander, J., Lundén, B., & Gavel, H. (2009). A computerized optimization framework for the morphological matrix applied to aircraft conceptual design. *Computer-Aided Design*, 41(3), 187–196. <https://doi.org/10.1016/j.cad.2008.06.005>
- OpenJS. (n.d.). *Electron* (9.4.4). <https://www.electronjs.org/>
- Pahl, G., Beitz, W., Feldhusen, J., & Harriman, R. A. (2007). *Engineering Design: A Systematic Approach; Third Edition*. In *Springer* (Issue 2).
- Ullman, D. G. (1992). *The mechanical design process* (Vol. 2). McGraw-Hill New York.
- Ulrich, K. T., & Eppinger, S. D. (2020). *Product design and development*. McGraw-Hill/Irwin.
- Weber, R. G., & Condoor, S. S. (1998). Conceptual design using a synergistically compatible morphological matrix. *FIE '98. 28th Annual Frontiers in Education Conference. Moving from "Teacher-Centered" to "Learner-Centered" Education. Conference Proceedings (Cat. No.98CH36214)*, 1, 171–176 vol.1. <https://doi.org/10.1109/FIE.1998.736828>
- Zwicky, F. (1967). The Morphological Approach to Discovery, Invention, Research and Construction. In F. Zwicky & A. G. Wilson (Eds.), *New Methods of Thought and Procedure* (pp. 273–297). Springer Berlin Heidelberg.