Problem examination for AI methods in conceptual product design

Philipp Rosenthal, **Oliver Niggemann**

Professorship Computer Science in Mechanical Engineering, Helmut Schmidt University
{philipp.rosenthal, oliver.niggemann}@hsu-hh.de

Abstract

Artificial Intelligence (AI) has significant potential for product design: AI can check technical and non-technical constraints on products, it can support a quick design of new product variants and new AI methods may also support creativity. But currently product design and AI are separate communities fostering different terms and theories. This makes a mapping of AI approaches to product design needs difficult and prevents new solutions.

As a solution, this paper first clarifies important terms and concepts for the interdisciplinary domain of AI methods in product design. A key contribution of this paper is a new classification of design problems using the four characteristics decomposability, interdependencies, innovation and creativity. Definitions of these concepts are given where they are lacking. Early mappings of these concepts to AI solutions are sketched and verified using design examples. The importance of creativity in product design and a corresponding gap in AI is pointed out for future research.

1 Motivation

Driven by advances in computer technology, Artificial Intelligence (AI) is becoming more important in recent years [BRD, 2020]. This is also true for the domain of product design and development. The design of a product determines how good a product can be manufactured, marketed and maintained throughout its life cycle. With shorter development cycles driven by globalization efficient designs and design processes become more important

AI can be helpful in several ways to support the design engineer. I can support by finding a vast number of solutions to a design problem as well as support in repetitive tasks like routine design. However, AI is not yet able to replace experience and creativity of human engineers. There is for example no general algorithm which solves the central problem of functional decomposition for any given design problem. This is in part due to the subjective nature of functional decomposition. Creativity is an important aspect for solving new design problems. So far all creativity in AI methods for solving design problems can be seen as induced by the programmer.

To identify potential AI applications in product design, the following research questions (RQ) must be answered:

RQ 1: Terms Which terms describe problem and solution approaches in design and AI and can they be harmonized?

RQ 2: Design/AI Mapping Can we define characteristics of product design problems, so that these characteristics correspond to the applicability of specific AI algorithms?

RQ 3: Evaluation Can we verify the new characteristics from RQ 2 by analyzing existing design examples?

The sections 2 and 3 address the fundamental terms in product design relevant to AI and the main directions AI has taken in in the design domain. Section 4 identifies problem characteristics. The following section takes examples from the the AI directions in product design and showcases the identified characteristics. The final section gives a summary, answers the research questions and gives an outlook for future research.

2 Important terms in product design

This section clarifies fundamental terms of product design and puts them into perspective for AI.

Product Design: Product Design (Product Development) is the purposeful evaluation and application of research results and experience. The aim is to obtain technical products, software programs, materials, principle solutions and similar concepts. [Verein Deutscher Ingenieure, 1993]

Design Synthesis: In regard to the design domain, the term synthesis can be seen from different points of view. Roozenburg [2002] gives a good overview over the different senses of the term. In general synthesis is understood as "putting something together" or as "collection". Analysis as the opposite of synthesis can be expressed as the resolution of objects into their elements. Synthesis is also a phase of the design process in which possible solutions for subproblems are found to build a complete design from. Here Analysis is the phase of identifying the subproblems. The VDI 2221 [Verein Deutscher Ingenieure, 1993] interpretes synthesis as a compilation of a list of designs which have to be evaluated to find the most suitable one for the given design problem. Synthesis as the assembly of subsystems by combining parts to a new whole and the functional analysis represented by e.g. [Pahl and Beitz, 1999] is also a prominent view on the term "synthesis". [Roozenburg, 2002]

Gagan et al. [2005] give a more computer centered description of synthesis in design. They state that design synthesis is the algorithmic creation of designs. A computer helps creating designs by organized, methodological modeling, implementation and execution.

Function and Behavior: The most important aspect of a product is its function. Chakrabarti [1998] defines function as intentional behavior. The intention is in this meaning distinct for being something actual or expected. E.g. wood swimming on water is in that regard a behavior, transporting goods on water is a function since it implies intention. Deng [2002] points out that function can be implemented with behavior. To continue the example, a raft can be build out of wood to transport goods.

Deng [2002] also makes an important distinction between purpose functions and action functions. A purpose function is an abstract and subjective description of intention of a product a designer has. He defines action functions as abstractions of intended and useful behavior. Purpose functions can be implemented via actions functions.

The third important concept of Deng [2002] is the distinction in function representation. Functions can be either semantically of syntactically represented. The latter one makes functions accessible to computers and is thus of interest for AI.

3 Classification of computational design synthesis

Chakrabarti et al. [2011] point out three major directions AI has taken in product design. Which are functional-based synthesis, grammar-based synthesis and analogy-based design. All three are addressed in the following.

Functional-Based Synthesis: The process of this approach is functional decomposition in conjunction with finding partial solutions to these subfunctions. The subsequent synthesis is the most crucial part since it defines the total function the product needs to fulfill in order to satisfy the desired requirements.

Chakrabarti and Bligh [1994; 1996b; 1996a] propose an approach to mechanical conceptual design using functional synthesis.

Grammar-Based Synthesis: Grammatical Design consists of three parts. First there needs to be a vocabulary of elements. The second part is a set for transformation rules and the final needed part is an initial structure to which the transformation rules can be applied [Brown, 1997]. The just mentioned grammar rules can be described as LHS →RHS. Where the left hand side (LHS) resembles the location where the rule is applied in the design and the right hand side (RHS) defines the design transformation. [Königseder and Shea, 2014]

Grammatical design is also known as design language. Design grammars are able to generate a wide variety of alternative designs to a design (sub-)problem. This generation is an important aspect of classical conceptual product design where the best evaluated solution is picked from a variety of design proposals.

According to Chakrabarti et al. [2011] the two most prevalent grammars in design are graph and spatial grammars.

There are six main steps which should be followed when engineering a design grammar. At first the representation has to be set, then the vocabulary needs to be defined. The third step is defining the grammar rules, followed by defining the initial design. The fifth step is the actual generation of design, followed by the final step of modifying the design grammar.

Analogy Design: Gero [2000] classifies analogy (among with combination, transformation, emergence and first principles) as a process for creative designing. He views analogy as a product of processes in which specific aspects of one problem are matched and transferred to another problem.

This is inline with Chakrabarti et al. [2011] who differentiate analogy based design further into case based design (CBD) and biology inspired design (BID). CBD is a type of case-based reasoning (CBR). Here, the case which is most similar to the new target problem is retrieved from a case-base. This case then helps to solve the new problem. [Kolodner, 1992] The similarity is crucial to CBR and can be expressed by a similarity function which maps the distance between the target problem and the cases in the case-base to a real number. The retrieved case is then reused to solve the given problem. In addition the solution is then revised to better fullfill any special requirements given by the problem. Ideally the new solution is then retained as a case in the case-base for future problems. [Hüllermeier, 2007]

BID can be used in the same way where the case-base consists of biological solutions to design problems. Databases containing biology inspired solution may also support functional-based synthesis.

An example for CBD would be the design of a truck by using analogical knowledge about an already designed car. BID has a prominent example in the development of airplanes. The necessary lift of objects was inspired by birds and how they create lift to fly.

4 Characteristics of Design Problems

A design problem is mainly characterized by the product-to-be-designed. The following subsections use product characteristics to characterize design problems. The main goal is to identify an alignment between these characteristics and suitable AI algorithms, this will be described in section 5 using exemplary use cases (RQ 2).

4.1 Characteristic 1: Decomposability

A popular classification for design problems is complexity. One may say that designing a car is more complex than designing a bicycle for instance. This may be an obvious observation when comparing problems from the same or similar class of design, in the above case personal transportation. However, it becomes vague when comparing problems from different classes of design. E.g. is a bicycle more complex than a bridge? A rational start is examining the function structure of a given design problem. In order to do that the overall function of the desired product must be accessible for functional decomposition.

According to Roozenburg [2002] this is not always the case. When considering spatial relationships, systems in

which the parts form an organic whole, any change in a sub-system providing a subfunction may impair the overall function. He calls these systems associate systems in contrast to flow systems, which are decomposable.

If a design problem and with it its function is very trivial, decomposition might also not be possible, simply because there exists no subfunction to the overall function. If a device has to be designed whose sole purpose is to absorb tensions, then there is no meaningful subfunction to it.

This leads to the binary dimension of decomposability. A design problem is either decomposable or it is not. Examples for the former are the aforementioned transportation devices like bicycles and cars. A representative of the latter is a bridge or rope.

4.2 Characteristic 2: Function Interdependencies

If a design problem falls into the decomposability category, a function structure can be generated. This structure is usually represented by a directed acyclic graph. An example for such a graph is shown in figure 1.
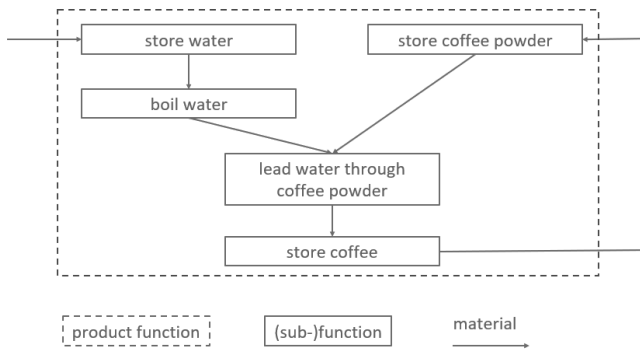

Figure 1: function structure - Here the overall function of *make coffee* is decomposed into several subfunctions. The subfunction *lead water through coffee powder* has an in-degree of 2 (boiled water and coffee powder) and an out-degree of 1 (coffee)

A large number of vertices or edges alone is not a good measurement for problem complexity, instead complexity is mainly causes by interdependencies between functions. And furthermore, a mere series of linear interdependencies, i.e. one predecessors and one successor, has a much smaller complexity than functions with high in-degrees or out-degrees. As a result the (nonlinear) interdependencies of a problem can be given as:

$$PI = \frac{number\ of\ vertices\ with\ a\ degree\ >\ 2}{number\ of\ all\ vertices} \quad (1)$$

Where PI denotes the (nonlinear) design problem interdependencies. This also leads to two conclusions about nondecomposable problems. First, if such a problem has multiple inputs and/or outputs, its PI equals one. This is because the black box representing the overall function is a single vertex. Figure 2 gives an example for such a black box.

Figure 2: Bridge as a Black Box - Here the overall function of *bear (dynamic) force* is not decomposed. The function has an in-degree of 2 (weight force and wind force) and an out-degree of 2 (diverted normal force and diverted transverse forces)

Second, if a nondecompasable problem exhibits only a single input and a single output, its PI will be zero. Such problems usually represent fundamental functions like the rope mentioned above.

4.3 Characteristic 3: Novelty of Design Problems

Intuitively, re-designing a product is easier than inventing something new. To formalize the term "new", Gero [2000] divided the design process into several mutual exclusive categories. These types can be used to describe how "new" a design problem is. A first distinction can be made between routine and nonroutine design. Routine design includes all design problems for which all necessary knowledge needed to find a solution is known a priori. Nonroutine design is distinct to that. Here not all knowledge needed to find a solution for the design problem is known right away. In addition, nonroutine design can be further devided into innovative design and creative design. [Gero, 2000]

To formalize these fuzzy notions, some definitions are needed:
Variables: Design knowledge is modeled based on variables. Each variable describes an, often physical, property of the product to-be-designed. Variables are often grouped into subfunctions. E.g. when a subfunction correspond to lifting an object, the corresponding variable may describe properties of a rotor such as size and amount of rotors.
Design Knowledge: Design knowledge are constraints on the variables, e.g. given in form of (in)equalities or in form of rules.

According to Gero [2000] Innovation in design occurs when unexpected values for variables in design become possible. An example for innovative design is to deduce a quadrocopter from a helicopter, the latter has an amount of one device that fulfills the function of applying lift to the object - a rotor in this case. While also having one device to counteract the resulting torque to the rotor axis. A quadrocopter is innovative because it extends the number of lifting devices beyond the prior known constraint and also reduces the number of devices needed to apply horizontal force. Analogical to PI a measurement for innovation can be given by:

$$I = \frac{number\ of\ variables\ with\ unexpected\ values}{number\ of\ all\ variables} \quad (2)$$

Creative design is even a step beyond innovation. Here the variables are (partially) unknown a priori and new variables are introduced into the design [Gero, 2000]. Before Guglielmo Marconi's work on radio transmission, signals could only be transmitted by wire or visually. The introduction of radio waves can be seen as a creative solution to the

problem of information transmission. Here the subfunction comprises a new variable "radio wave". As with innovation, a computational measurement for creativity can be given by:

$$C = \frac{number\ of\ new\ variables}{number\ of\ all\ variables} \qquad (3)$$

Boden [2009] points out a few more important aspects of creativity which should also be applied to innovation. An important constraint to creativity is that creative solutions have to be valuable. Not all alterations of values or variables result in a feasible solution. If that were the case, innovation and creativity would be trivial concepts which could be implemented by a randomizer. The tricky part is judging which alterations are imbued with value.

The other aspect Boden [2009] mentions is the range of creativity. It can be distinguished between psychological creativity (p-creativity) and historical creativity (h-creativity). The former applies when the solution is new to the individual that created it and the latter applies when in addition the solution is new to the world.

Once unknown values or new variables are introduced into a class of design problems, these become known knowledge. In conclusion innovative and creative design can only be examined in retrospective where the innovation and / or creativity was unknown.

5 Use Cases and Mapping to AI Algorithms

This section takes examples representing the aforementioned AI directions in product design from research and incorporates them into the described problem classification.

5.1 Grammar for Shaft Design

To demonstrate the mode of operation for design grammars Brown [1997] introduced the problem of designing a stepped grooved shaft. the overall function of such shafts is to forward torque while also diverting axial and radial forces. From this description a decomposability into basic subfunctions lies at hand.

Classification into Categories: The example of a shaft is not complex, since subfunctions mostly have only one input and one output, e.g. torque. The resulting PI equals zero. Even though the sections may also serve for e.g. bearing forces, there is no complexity since the shaft itself does not create interdependencies. For the vocabulary for this particular example, Brown states two shapes, being grooved and ungrooved and two labels and five simple transformation rules. Innovation occurs, when variables, in this case, section length, diameter and number of sections have unexpected values. It is possible, that the given grammar produces designs for shafts with unknown values for variables but their usefulness and with that their innovativeness can be argued. Since for design grammars themselves the vocabulary needs to be known a priori, creative solutions are impossible, since a design grammar is not able to introduce entirely new variables to a design, e.g. a square diameter section for a shaft which has no bearings.

Potential AI Approaches: In AI, such knowledge models can be captured e.g. using rules [Eastman *et al.*, 2009]. Nowa-
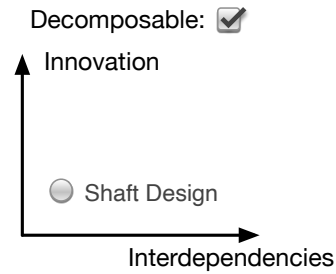

Figure 3: Categorization of the Shaft Design Example

days, often machine learning is used to either extract those rules [Ko *et al.*, 2021] from data.

5.2 Grammar for Gearboxes

Holder et al. [2019] introduce a design language to synthesise gearboxes including 3D component arrangement. The overall function of a gearbox is to transmit torque. Subfunctions needed to synthesise this function are e.g. bearing of shafts, transmissions or reversal of torque direction.

Classification into Categories: As a result the problem of designing a gearbox is decomposable due to its identifiable and separable subfunctions. Responsible for the PI of gearboxes are functions which require to have radial or axial forces along with torque as input and corresponding outputs like diverting forces. Such a function may be represented by a bearing. When also considering 3D placement of parts the PI grow due to constraints which may add subfunctions to the design problem. E.g. a very narrow installation place may lead to additional gearwheels to achieve the required transmission. As with the shaft example, innovation may be achieved by an unexpected value for the given variables, e.g. number of gearwheels or the transmission between gearwheels. Depending on the given vocabulary of the design grammar an additional change in torque direction is also thinkable. Also creativity is not possible to achieve since all possible variables have to be included into the knowledge base being the vocabulary of the design language.
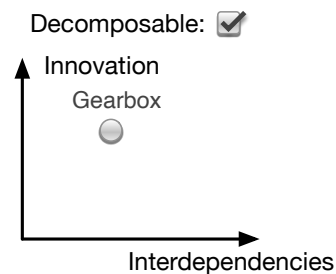

Figure 4: Categorization of the Gearbox Example

Potential AI Approaches: Such grammars [Schmidt and Cagan, 1996] are another AI approach which is an extension of rule based approaches. By defining an order of symbols, geometric and other dependencies between subsystems can be

modeled. Furthermore, their generative nature fits well to design [Königseder and Shea, 2014]. In the last years, learning of grammars has become more and more popular [Gero *et al.*, 1994].

5.3 Analogy Based Coil Winder Design

McAdams and Wood [2002] approached analogical design using a similarity metric. They chose a coil winder for guitars as a problem to test the analogical design. This problem is well suited to demonstrate the characteristics of design problems which can be solved using the analogy approach.

Classification into Categories: First the design problem is decomposable. Figure 5 depicts the function structure of such a device. From the function structure the PI measurement can be derived. In the shown example 12 of 28 nodes have a degree greater than two, in conclusion the PI has a value of $3/7$.
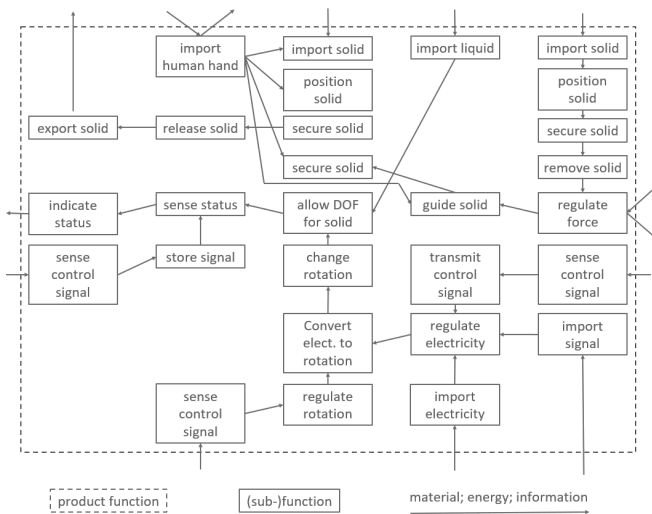
Figure 5: simplified Coil Winder function structure based on [McAdams and Wood, 2002]

Analogy based design solutions are open to creativity, as long as the analogies be it from the problem domain or not are included in the knowledge base. This implementation is usually done by the programmer. In conclusion the creativity does not originate in the computer model but in the person programming it. This is the case in the coil winder example, where a fruit peeler and a fishing reel are included in the knowledge base among other products. According to the example, using prongs to secure the bobbin can be seen as creativity originating in the experience from the fruit peeler. Innovation is possible in the same way creativity is using analogy design techniques. However, the example given by McAdams et al. does not include any innovative variable settings.

Potential AI Approaches: In AI, case-based reasoning has always been a successful approach to analogy-based design [Goel and Craw, 2005; Maher and Pu, 1997]: First, a function is defined which captures the similarity between designs. Then for a new design, a similar old design is identified
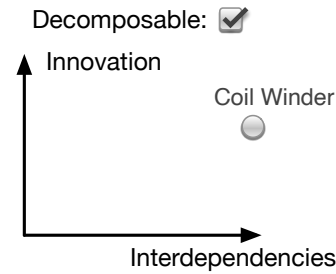
Figure 6: Categorization of the Coil Winder Example

and adapted to the new requirements. In the last years, generative neural networks offer a new analogy-based solution approach [Kusiak, 2020; Wang *et al.*, 2019]: A neural network uses examples to learn the mapping from requirements to product designs and is able to use this learned knowledge to generate new designs for new requirements.

5.4 Circuit Design

Feldman et al. [2019] proposed an approach for Boolean circuit design, where circuits are generated using quantified Boolean formulas. Their algorithms solved two proposed problems. While the first problem was generating a Boolean circuit given a circuit topology and the desired requirements, the second problem also included the automatic generation of the circuit topology.

Classification into Categories: The approach falls in the category of functional-based synthesis. Considering the decomposability of boolean circuits, it becomes clear that the topology of these circuits represents a function structure. These structures can be decomposed into subfunctions which are constituted by circuit components like, e.g. gates. Aside from determining the decomposability, the PI are also computed by looking at the topology. Each gate can be seen as a vertex. Here vertices have either one input and one output, e.g. identity and negation, or two inputs and one output, e.g. AND, OR, XOR gates. Given a topology the PI can be computed using equation 1. A full substractor is shown in figure 7. The PI amounts to $5/7$ in this example.
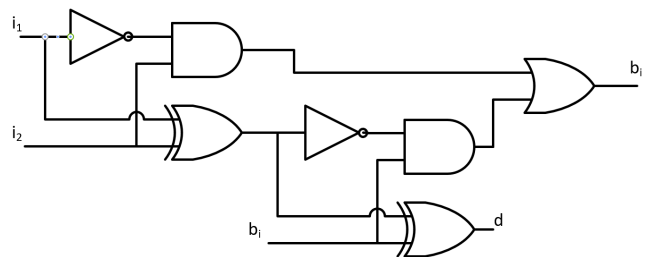
Figure 7: Full substractor based on [Feldman *et al.*, 2019]

The proposed approach can be seen as innovative, when one considers that the algorithms used are able to find solutions which require a different number of components and different compositions. Here the values for the given vari-

ables, i.e. number of certain components, may be shifted towards prior unknown values. Which gives the approach innovative capabilities. Since the algorithms are not able to find new variables i.e. the logic gates used to build Boolean circuits, the approach shown by Feldman et al. is not capable of being creative.
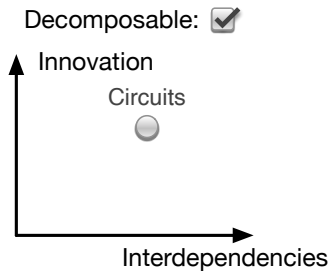
Decomposable: ☑

Figure 8: Categorization of the Circuit Example

Potential AI Approaches: In AI, such combinatorial problems are often solved using search algorithms such as constraint systems [Bowen *et al.*, 1993] or genetic search [Krestinskaya *et al.*, 2020]. Such solutions are helpful if the search space structure is well known.

6 Conclusion

Sections 2 and 3 answer RQ 1. Further, design problems can be categorized by their decomposability. Closely tied to decomposability are the problem interdependencies. Meaningful statements about the PI can only be made if the problem is decomposable into a function structure. The two exceptions are trivial design problems which can be solved with fundamental concepts and nondecomposable problems, for which no PI statement can be made apart from viewing the problem as a black box. The remaining two characteristics are concerned with the novelty of nonroutine design problems. While innovation occurs by altering variables to unknown values, creativity goes even further by implementing new variables. These characteristics correspond to RQ 2.

In relation to RQ 3, the exemplary use cases from the three main directions of AI in product design point out that problems need to have specific characteristics in order to be tackled by these methods. Figure 9 uses the new categories to classify AI technologies for product design. The analysis is based on the examples where each example has been analysed in terms of characteristics and AI solutions.

As long as a problem is decomposable, functional synthesis, design grammars and analogy design approaches are all suited for design problem solving. All approaches are suited to deal with problem interdependencies, given the problem can be mapped to a function structure.

Innovation may be achieved with all mentioned approaches. Altering variable values can be easily done by design grammars by implementing transformation rules which allow bypassing or resetting constraints. In functional-based synthesis innovation is also possible by altering values of variables, which the circuit design example showed. Anal-
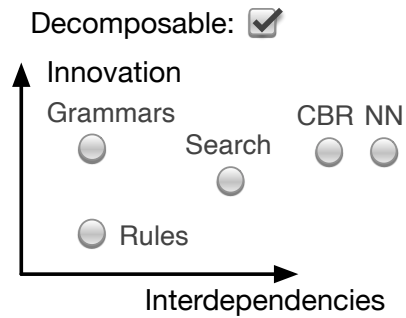
Decomposable: ☑

Figure 9: Mapping AI methods to design problem categories.

ogy approaches are less suited for innovation. Here the similarity measurement puts the solutions as close as possible to already existing ones. This leaves less room for assigning unknown values to existing variables. For all approaches there needs to be some kind of measurement for when the alteration led to value. This may be done with a cost function, e.g. less components lead to cheaper production and easier maintenance. Interestingly none of the approaches is able to implement creativity.

This answers the question asking for problem characteristics which are not handled sufficiently by AI methods. The main reason these methods cannot handle creativity is that all approaches refer to a knowledge base which is externally implemented and assumed to be suited best to the given problems by the programmer. Design grammars are limited to their vocabulary, functional-based approaches rely on a knowledge base and analogy approaches refer to an analogy or case base.

Creativity in product design is what is responsible for disruptive advances, be it the development of machines, software, the internet or even AI itself. Giving the computer the ability to find creative solutions to design problems would be of great assistance for finding solutions for novel design problems. Being able to produce creative solution on a reliable base would greatly improve the product design process and save costs in the development of novel products. Researching creativity could start out on the psychological level to prove new concepts. The ultimate goal should be to achieve historical creativity.

References

[Boden, 2009] Margaret A. Boden. Computer models of creativity. *AI Magazine*, 30(3):23, 2009.

[Bowen *et al.*, 1993] J. Bowen, D. Bahler, and A. Dholakia. An ai constraint network-based approach to bed-of-nails dft for digital circuit design. *Computers and Electrical Engineering*, 19(2):73–85, 1993.

[BRD, 2020] BRD. Strategie künstliche intelligenz der bundesregierung fortschreibung 2020. PDF, 2020.

[Brown, 1997] K. Brown. Grammatical design. *IEEE Expert*, 12(2):27–33, 1997.

[Cagan *et al.*, 2005] Jonathan Cagan, Matthew I. Campbell, Susan Finger, and Tetsuo Tomiyama. A framework for

computational design synthesis: Model and applications. *Journal of Computing and Information Science in Engineering*, 5(3):171–181, 2005.

[Chakrabarti and Bligh, 1994] Amaresh Chakrabarti and Thomas P. Bligh. An approach to functional synthesis of solutions in mechanical conceptual design. part i: Introduction and knowledge representation. *Research in Engineering Design*, 6(3):127–141, 1994.

[Chakrabarti and Bligh, 1996a] Amaresh Chakrabarti and Thomas P. Bligh. An approach to functional synthesis of solutions in mechanical conceptual design. part ii: Kind synthesis. *Research in Engineering Design*, 8(1):52–62, 1996.

[Chakrabarti and Bligh, 1996b] Amaresh Chakrabarti and Thomas P. Bligh. An approach to functional synthesis of solutions in mechanical conceptual design. part iii: Spatial configuration. *Research in Engineering Design*, 8(2):116–124, 1996.

[Chakrabarti *et al.*, 2011] Amaresh Chakrabarti, Kristina Shea, Robert Stone, Jonathan Cagan, Matthew Campbell, Noe Vargas Hernandez, and Kristin L. Wood. Computer-based design synthesis research: An overview. *Journal of Computing and Information Science in Engineering*, 11(2), 2011.

[Chakrabarti, 1998] Amaresh Chakrabarti. Supporting two views of function in mechanical designs. *Workshop on Functional Modelling and Teleological Reasoning, Fifteenth AAAI National Conference on Artificial Intelligence, Wisconsin*, 1998.

[Deng, 2002] Y.-M. Deng. Function and behavior representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 16(5):343–362, 2002.

[Eastman *et al.*, 2009] Charles M. Eastman, Jae-min Lee, Yeon-suk Jeong, and Jin-kook Lee. Automatic rule-based checking of building designs. *Automation in Construction*, 18(8):1011–1033, 2009.

[Feldman *et al.*, 2019] Alexander Feldman, Johan de Kleer, and Ion Matei. Design space exploration as quantified satisfaction. 2019.

[Gero *et al.*, 1994] J. Gero, S. Louis, and S. Kundu. Evolutionary learning of novel grammars for design improvement. *Artif. Intell. Eng. Des. Anal. Manuf.*, 8:83–94, 1994.

[Gero, 2000] John S. Gero. Computational models of innovative and creative design processes. *Technological Forecasting and Social Change*, 64(2-3):183–196, 2000.

[Goel and Craw, 2005] Ashok K. Goel and Susan Craw. Design, innovation and case-based reasoning. *Knowledge engineering review*, 20:271–276, 2005.

[Holder *et al.*, 2019] Kevin Holder, Stephan Rudolph, Ralf Stetter, and Corinna Salander. Automated requirements-driven design synthesis of gearboxes with graph-based design languages using state of the art tools. *Forschung im Ingenieurwesen*, 83(3):655–668, 2019.

[Hüllermeier, 2007] Eyke Hüllermeier. *Case-Based Approximate Reasoning*, volume 44 of *Theory and Decision Library*. Springer, Dordrecht and Berlin and Heidelberg, 2007.

[Ko *et al.*, 2021] Hyunwoong Ko, Paul Witherell, Yan Lu, Samyeon Kim, and David W. Rosen. Machine learning and knowledge graph based design rule construction for additive manufacturing. *Additive Manufacturing*, 37:101620, 2021.

[Kolodner, 1992] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3–34, 1992.

[Königseder and Shea, 2014] Corinna Königseder and Kristina Shea. Systematic rule analysis of generative design grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 28(3):227–238, 2014.

[Krestinskaya *et al.*, 2020] Olga Krestinskaya, Khaled N. Salama, and Alex P. James. Automating analogue ai chip design with genetic search. *Advanced Intelligent Systems*, 2(8):2000075, 2020.

[Kusiak, 2020] Andrew Kusiak. Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research*, 58(5):1594–1604, 2020.

[Maher and Pu, 1997] Mary Lou Maher and Pearl Pu, editors. *Issues and Applications of Cased-Based Reasoning in Design*. Psychology Press, New York and London, 1997.

[McAdams and Wood, 2002] Daniel A. McAdams and Kristin L. Wood. A quantitative similarity metric for design-by-analogy. *Journal of Mechanical Design*, 124(2):173–182, 2002.

[Pahl and Beitz, 1999] Gerhard Pahl and Wolfgang Beitz. *Engineering design: A systematic approach*. Springer, London, 2. rev. ed., 2. print edition, 1999.

[Roozenburg, 2002] Norbert F. M. Roozenburg. Defining synthesis: on the senses and the logic of design synthesis. In Amaresh Chakrabarti, editor, *Engineering Design Synthesis*, pages 3–18. Springer, London, 2002.

[Schmidt and Cagan, 1996] Linda C. Schmidt and Jonathan Cagan. Configuration Design: An Integrated Approach Using Grammars. Volume 4: 8th International Conference on Design Theory and Methodology, 08 1996.

[Verein Deutscher Ingenieure, 1993] Verein Deutscher Ingenieure. Vdi-richtlinie 2221: Methodik zum entwickeln und konstruieren technischer systeme und produkte, 1993.

[Wang *et al.*, 2019] Guodong Wang, Anna Ledwoch, Ramin M. Hasani, Radu Grosu, and Alexandra Brintrup. A generative neural network model for the quality prediction of work in progress products. *Applied Soft Computing*, 85:105683, 2019.