Towards Synthetic Engineers: Requirements & Implications of the Conceptual Engineering Design Process

Chloe A. Schaff[1]✉ and Kristinn R. Thórisson[1,2]

[1] Center for Analysis & Design of Intelligent Agents (CADIA)
Reykjavík University, Iceland {chloes21,thorisson}@ru.is

[2] Icelandic Institute for Intelligent Machines (IIIM)
Reykjavík, Iceland

Abstract. Human design of any novel system or artifact, e.g. a new type of vehicle, house, or satellite, rests on both general real-world engineering knowledge and inventiveness. At the outset of the conceptual stage of any such design process, requirements are often vague and conflicting—even missing. Consequently, the conceptual engineering design process has proven difficult to implement in machines. Thus, prior work on design automation has not surprisingly focused largely on the later steps of design, which tend to be much more structured. We envision a future of "synthetic engineers" that can help human experts with increasingly complex and challenging systems design. Naturally, we may ask what contemporary research on artificial general intelligence could contribute to this vision. Conversely, we can investigate what goes on in the conceptual engineering design process and ask whether this may provide valuable insights into research on general machine intelligence. Taking this perspective, and based on a review of two existing implemented cognitive architectures, we present a set of what we consider necessary cognitive faculties that must be *coherently unified in a single agent architecture* to automate the conceptual design process, and a set of minimum requirements that any agent capable of conceptual design must meet.

Keywords: Artificial Intelligence · Engineering Design · Dynamic Planning · Defeasible Non-Axiomatic Logic · Cumulative Learning

1 Introduction

Engineering design is a complex process that requires reconciling many design requirements at varying levels of abstraction and detail, some of which may conflict, be poorly understood, or be in flux during the process [19]. We define a *design problem* as the description of an as-yet unaddressed need that may be met by the development of a new structure, process, or arrangement of components and/or elements. *Engineering design*, in turn, is the process of finding optimal solutions to the stated requirements given available resources. Successfully addressing an open design problem requires general problem-solving skills as well as physical action. To be successful, a solution to a design problem must reconcile all stated requirements (constraints), which typically involve descriptions spanning multiple levels of detail.

The process of solving an engineering problem requires iteration over questions and solutions, adapting to new conclusions, thinking creatively, and drawing on experience. These are all skills that are difficult to capture in software systems and, while computers have been used to assist human engineers for some time [4, 14, 16], very few have aimed to reach higher levels of automation in a way that allows the computer to actively participate in – or perhaps even direct – the *conceptual* problem-solving phase of the process. Framing this discussion are Altavilla and Blanco's [2] levels of automation: at Level 1 the human designer directs and executes everything, at Level 5 this is all performed autonomously by a computer, and in between is a range wherein the computer provides varying degrees of assistance to a human designer. Existing design automation systems stand somewhere around Levels 2 to 3, in which human designers solve the problems while computers mainly help with the execution. In order to reach Levels 4 or 5, we propose applying existing research on general machine intelligence (GMI) which has spent decades developing systems with exactly the skills required to solve more advanced and conceptual problems. Additionally, better automation of conceptual engineering is likely to have significant practical importance, since enhancing humanities' ability to solve engineering problems would be of notable value in virtually every industry. Systems operating at Levels 4–5 in the automation hierarchy would be able to assist in problem-solving and, covering a larger part of design process, empower human engineers by allowing them to focus on the higher-level and more creative parts of the problem.

We believe that some initial efforts in this direction can be made with existing GMI-aspiring cognitive architectures. Here we look at three key challenges in the design process that such systems must address, *autonomy*, *creativity*, and *causality*, and propose a set of minimum requirements for these that any agent capable of high-level conceptual design automation must meet. By seeking a better understanding of the design process and establishing basic system requirements, the work aims to pave the way towards the first synthetic engineers.

2 Related Work

Though engineering design has been the subject of much research, it would appear that most theoretical conceptions of the design process are human-centric and that no formal rigorous definition of the engineering method exists yet [17]. This is not to say that *no* attempts at formalizations exist, however. A good example is the field of 'axiomatic design' pioneered by Suh in 1977 [14], which organizes design around an 'independence axiom' that keeps designs modular and an 'information axiom' that keeps them simple. These are applied through, among other techniques, a design matrix that maps design requirements to design outcomes (referred to as 'functional requirements' and 'design parameters,' respectively) and analyzes any resulting dependencies of the outcome.

A similar approach can be seen in TRIZ [16], which proposes several guiding laws of engineering design. Formalizations that directly consider the reasoning involved in the design process also exist; an example is the SAPPhIRE model [4], which tracks the progression of design from abstract high-level processes down to the details of implementation. Finally, Žavbi and Duhovnik [30] offer a particularly interesting perspective by approaching a design as a series of linked causal models based on physical laws. They demonstrate how, for example, a microphone can be invented by chaining an equation for acoustic energy to one for capacitance, then to one for an RC circuit, and then finally to an equation for electric current. All of this work demonstrates that engineering design involves processes that *can* be formalized, though not all of this work has yet been integrated into a single software system.

Existing developments in practical design automation systems tend to focus on lower levels of automation and broadly belong to one of two categories: *numerical* and *reasoning*-based. Numerical systems directly manipulate geometry to generate novel structures and solutions to problems; a key field here is generative design (GD) [10] which applies genetic algorithms, swarm optimization [6] and related methods to find a solution to a set of physical design constraints. Reasoning-based approaches can greatly accelerate engineering design given the potential to facilitate a *guided search* for a solution and generalize good solutions onto a wider variety of problems. An example of existing reasoning-based approaches to design is in knowledge-based engineering (KBE) systems which use pre-planned causal relationships to capture human knowledge and efficiently develop designs for mass-customized products [26]. Finally, also worth mentioning are large language models (LLMs) which initially appear to fall somewhere in between numerical and reasoning-based. It should be noted that, while LLMs have demonstrated some engineering capabilities [9] and may be useful for human designers in their ability to pose as potential users of a product [11], they do not fundamentally represent causal relationships and so what reasoning they do perform is only approximate at best [29]. Despite all of these developments and regardless of architecture, it does not appear that any existing engineering systems are capable of operating with the level of autonomy and cumulative learning of a human engineer: numerical approaches do not have the ability to learn from experience or consciously extract high-level patterns in their work, knowledge-based systems can only handle so much novelty before a new model must be constructed by a human engineer, and LLMs often struggle with conceptualizing the problem, staying on task, and must be prompted continuously and very carefully. Systems capable of a higher-level of design automation will need to be able to confront these challenges and this may require an entirely new cognitive architecture.

Many AI researchers consider contemporary AI methodologies to be inadequate for reaching general machine intelligence [1]. The Non-Axiomatic Reasoning System (NARS) [27] and the Autonomous Empirical Reasoning Architecture (also called the Autocatalytic Endogenous Reflective Architecture or AERA[3])

[3] See http://www.openaera.org – *accessed May 9th, 2025.*

[13] are two GMI-aspiring systems developed with a strong focus on reasoning and experiential learning; AERA with a particular emphasis on the kind of causal reasoning [25] discussed by Žavbi and Duhovnik [30] and Campbell et al. [5]. These systems have recently been applied in an engineering context [17]; there is good reason to investigate both further for their potential to solve known limitations of current approaches to conceptual design automation.

3 Challenges in the Conceptual Design Process

We define three key challenge areas with respect to autonomous conceptual engineering design: causality, creativity, and autonomy. First, the concept of *modeled causality* provides a necessary framework for effectively and efficiently representing design constraints and the relationships between them; it is this structure that gives rise to explainable design [23], a feature essential to any professional engineering design solution. *Creativity* is posited as an essential mechanism by which under-constrained problems can be solved and innovative solutions generated and explored [20]. Finally, the concept of *autonomy* captures how an engineer must be free to make their own decisions as both problems and solutions change and grow throughout the design process [22].

3.1 Causality in the Conceptual Design Process

When starting on a real-world engineering design, there is rarely the time, space, or energy to explore every potentially-relevant small detail for every possible high-level design option, and attempting to do so may threaten the completion of the design process as a whole.[4] For simple problems, an exhaustive exploration may be feasible, but in order to save time and resources, we expect most problems to require designers to be able to abstract away from low-level details and prioritize effectively. Imagine a worst-case scenario wherein all details are considered relevant: in this case the high-level design could not proceed until all questions about these minutiae have been answered—the time for completing the conceptual design is equal to that for completing the whole design. Needless to say, this approach make some complex designs (e.g. bridges, aircraft, smartphones, etc.) computationally intractable. Human designers constrain their search space by, among other things, decomposing the problem's requirements and applying their knowledge of the problem domain to these requirements. The most obvious way to do this is through general, and sometimes specific, cause-effect laws, that preclude large swaths of potential options from practical consideration, thereby making the task more feasible. A human designer can then explain when, where, why, and how they abstracted a problem, and justify such decisions with verifiable arguments based on valid and verifiable cause-effect relations [25].

[4] There is no option here for ignoring time in any practical automation of engineering design, as any such system must be implemented on hardware running in the physical world. Hence, all plans in the physical world are inherently time-constrained.

This may not be as hard as it seems; design problems always come with a set of requirements (with a goal to solve a specific problem in a particular way within a limited time; c.f. [3, 17, 24]) which can often be (eventually) grounded in knowledge of the physical world (assuming the problem is solvable). Additionally, very few design requirements are fully independent, and relationships between requirements often point in the direction of possible solutions. In this way, the causal relationships between requirements, domain knowledge, and other potential candidate solutions act not only to constrain the search space but can actually guide the design.[5] Consider a water bottle as an example: With the requirements "must contain 1 liter" and "must be comfortable to hold" we can see that, while 1 liter of water could take any shape, only certain shapes in certain sizes are comfortable to hold in the average human hand; one requirement interacts with another so as to constrain and guide towards a solution.

During conceptual design, many potential solutions that seem viable may actually be in conflict. For instance, to satisfy the "must be comfortable to hold" requirement, it might seem perfectly fine to put a handle on the water bottle, based on available prior examples. However, guiding the design work towards an acceptable solution requires a representations that can, in this case, produce information about incompatible relationships between handle size and small-bag holding capacities. Without knowledge that allows answering questions about *why* a solution may (or may not) satisfy requirements, separation between solutions that are in conflict and those that are not may be impossible. Knowledge enabling production/analysis of the details in the mapping between requirements and solutions is needed,[6] and this must involve causal relations [15].

Ultimately, to trust a design blueprint, we must be able to get answers about how it works; the ability to argue using causal relationships between requirements and solutions is essential for establishing this trust. Reasoning over causal relations is also needed for explanation generation; in a particularly intentional design, every engineering design decision must come with an argument against possible alternatives – "Why this option over that option?" – and an argued relationship to the design requirements—"Why is this needed at all?". Consider the water bottle again; with a design based on causal relations one could ask "Why is this water bottle a cylinder?" from which an explanation is produced that this is "because it contains 1 liter of water and is comfortable to hold," with appropriate arguments (e.g. a list of alternatives that demonstrably fail to meet requirements). Without reliable cause-effect models it is not clear at all how such a response could be generated. It clearly could not be done with a solely statistics-based approaches, as this could only offer suggestions about what is *likely* to work, based on past observations, that they will *almost certainly* work in this (possibly novel) circumstance, based on given background assumptions.

The intermediate conclusion here is this: an AI system capable of conceptual design must be able to discover causal relationships in design requirements and

[5] A particularly intentional design will be guided almost entirely by the relationships between its design requirements; see the previously-discussed example from [30].

[6] For a discussion on reasoned mappings versus statistical mappings, see [29].

then reason about these to constrain the search space. Only then can it propose a design that can then be *explained*, *argued for*, and eventually *tested* based on its causally-grounded reasoning.

3.2 Informed Creativity in the Design Process

A fundamental challenge of an under-constrained problem is that, rather than constraints pointing the way towards a unique solution, they open up to a multitude of possible solutions. Narrowing these down to a final candidate solution requires the designer to either discover more constraints or make an assumption or decision about how the solution should look based on their own judgment. One context in which creativity seems relevant in engineering design is in the handling of such under-constrained problems. We see ‘creativity’ here as the informed steering of making choices between alternatives, by selecting (from prior experience) or generating explicit custom arguments, according to some chosen principles, that can later be re-analyzed, reconsidered, and possibly reverted, in light of new information.

To discover more constraints, a designer could work within the constrained areas of the design and perform experiments to try to uncover them through experience; this is the domain of research and development. The designer could also attempt to further break down the existing constraints to elicit factors or relationships at a finer level of detail. If more constraints cannot be discovered, the designer can use their experience to substitute in solutions that have worked for similar problems in the past, make their own assumptions as to what they think the solution needs, or invent a wholly novel concept to fit the requirements in a new way. Slotting in similar solutions is relatively straightforward, though this does require the ability to learn from problems and retain successful strategies; these can narrow the search space by pointing the design in a certain direction that can be assumed to be promising based on prior experience. Making such assumptions requires some understanding of the problem itself [21]. As in the discussion of causality, the designer may use their own interpretations (read: causal models) of the problem domain and select a solution based on aesthetic principles or because it intuitively seems like the better option; if new constraints are discovered, this direction can always be changed. Finally, there may be a necessity to create a new concept from scratch; though this kind of innovation is a much more advanced technique. For instance, in a world without hinges, the concept of a “hinge” is a radical one—such ideas can only spring forth through a process of analogy through ampliative reasoning [18, 24]; these are key features of intelligence that are difficult to tease apart [22].

Regardless of how an under-constrained problem is resolved, it requires the engineer to creatively reason their way through the situation. If they are unable to further decompose the problem requirements, they must be able to make a creative decision as to which direction the problem-solving process should go. These decisions can be justified based on prior experience, assumptions, or innovation but they must be made in order to select a design (or set of designs) for further design, testing, and possible implementation [8]. All of this can be supported by the design system's ability to apply causal reasoning and argumentation to its design options, as this can allow it to explain and justify its creative decisions and better apply its knowledge to under-constrained problems that may require a dash of informed creativity.

3.3 Autonomy & Control of the Design Process

These final challenges are fundamentally about how a design system guides its problem-solving process. Like any human engineer, a synthetic design system exploring a massive set of possible solutions must be free to make its own informed decisions about how to analyze the problem constraints and in which direction to take current solution candidates.

In terms of requirements analysis, it is rare for a problem to be fully and comprehensively described at the outset of a design task; Smithers [19] posits that every design problem begins with an Initial Requirement Description (IRD), a description that is "incomplete, inconsistent, imprecise, or ambiguous or (more typically) some combination of all of these" [19, p.7]. The first step in the solution process is to gain a better understanding of what the requirements mean and what motivates them. Sometimes this can be done by asking questions to elicit the finer details of the requirements but it can also occur by suggesting possible solutions and getting feedback on why they may not satisfy the IRD. In this way, the problem specification and solution candidates tend to evolve together.

Consider, for instance, that a person decides they need a new computer. An IRD consisting solely of "I need a new computer" is clearly incomplete and the designer will need to flesh this out to gain a better understanding of the problem. Often this involves prioritizing requirements ("it must be portable", "it must be within my budget", "it should fit into my bag", "I would prefer a touchscreen", "the color does not matter", etc.). A key feature of this process is the need to understand the reasons for these requirements: "Why does the user need a new computer?", "What do they hope to do with it?", "Can that need be better serviced by a smartphone?". The exploratory nature of the design process can also occasionally reveal entirely different solutions that may actually satisfy the design specifications better than the obvious solution. However, none of this is possible unless the designer has the autonomy to explore the solution space under its own direction; it must be able to apply its causal understandings of the requirements (as grounded in the physical world) to suggest possible solutions to the IRD. Unpacking IRDs into constraints and meaningfully connecting them to the domain that gives the design its context leads to useful solutions. The decisions the designer makes to apply causal knowledge and reason creatively around problems must be autonomous.

In doing so, it must not be restricted by simple fixed notions of a given problem or even of the design process itself. Very few designs progress in a fixed feed-forward nature; iteration and reflection are an important part of the problem-solving process. This is where autonomy becomes critical in the design process: When a problem is incompletely specified, a designer cannot simply proceed linearly through the design process as if the solution were obvious from the outset. The design system must be able to control the direction of this process such that it can follow paths that seem promising, revert back to the drawing board when a dead end is discovered, and eventually select and propose a solution candidate when/if one is found. Doing all of this requires giving the system the necessary capacity for planning and autonomy so it can conduct its own exploration and commit to design decisions under its own direction. Without this, the human would still be left making all the key decisions.

4 Requirements for Conceptual Engineering Design

Based on these challenges, we can now summarize the key cognitive faculties that an AI system must posses to be capable of conceptual engineering design. We contend these are necessary – but possibly not sufficient – to meet the challenges of the conceptual design processes outlined above. The list makes it rather clear that these cognitive functions are co-dependent, and could not each be implemented as separate interacting systems; they must be seamlessly integrated under one cognitive architecture. Any such cognitive architecture, however, would have reached a strong starting point for addressing engineering design capabilities.

For conceptual engineering design, an agent must be able to:

§1 $\langle R \rangle$ Consider the causal (and other) relationships between individual requirements, as well as between the full set of requirements and any proposed solution. This would enable it to compare design features and requirements against real-world outcomes.

§2 $\langle C, R \rangle$ Make new categories and comparisons when existing knowledge is insufficient, exploring unconventional and completely new solutions in an informed manner.

§3 $\langle A \rangle$ Reason in an iterative fashion through the design, progressing from high levels to low levels of detail, through loops of re-consideration and re-design of prior levels and concepts. The agent should also be able to reason about the design process itself and identify the best ways to use its time for problem solving.

§4 $\langle C, R \rangle$ Deal with underconstrained problems at any level of detail by decomposing requirements, suggesting solutions, and innovating.

§5 $\langle A, C, R \rangle$ Not be constrained to a simple top-down or bottom-up approach. The design process is causal, creative, and fluid, so any designer must be able to autonomously progress through the design process.

Brackets $\langle\ \rangle$ indicate which of the three categories each requirement calls on — $\langle R \rangle$: Experience-grounded knowledge of physical causal relations; $\langle C \rangle$: Creativity; $\langle A \rangle$: Autonomy and control.

5 Cognitive Architectures & Engineering Design

As discussed in the Related Work section, most prior approaches to engineering design automation [10, 26] lack a sufficient capacity for autonomous reasoning, and thus tend to be limited to well-defined designs. Most of them also lack reasoning abilities [12]. Building a system with more advanced conceptual design capabilities will require a different cognitive architecture that we have seen to date. Two existing architectures that are promising for this purpose are NARS [27] and AERA [13]. NARS is designed from the ground up to reason about processes and facts, and AERA has been demonstrated to be capable of learning and reasoning from experience about causal relationships. Both systems can handle a problem and solution evolving simultaneously and are capable of discovering constraints as they work iteratively through a problem. These qualities make both systems of particular interest in engineering applications. As it stands, these systems meet the requirements given in Section 4—they should possess all the required cognitive faculties to handle design problems.

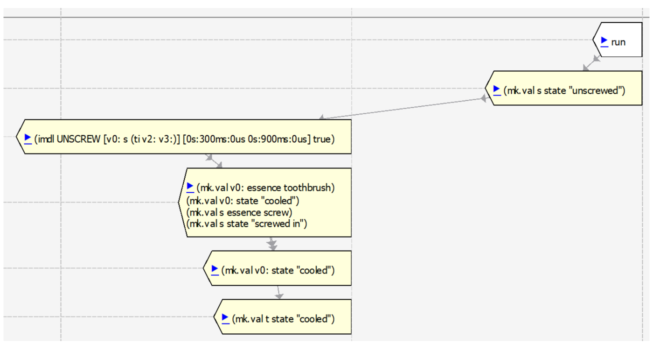

Fig. 1. The first steps of AERA's plan to unscrew a screw by designing a screwdriver made out of a heated, reshaped, and cooled toothbrush (reproduced from [17]; see also [28]).

While efforts have been somewhat limited towards implementing synthetic engineers in these architectures, some notable progress has been made in this direction. Here we will mention the design problem of unscrewing a screw without a screwdriver, as demonstrated in the OpenNARS for Applications project [7], using only a toothbrush and a lighter. The problem can be solved by way of fashioning a screwdriver out of the toothbrush using the lighter, by heating it up, thrusting it into the screw to make it deform to its imprint, waiting for the plastic to harden, and then using it to unscrew the screw. This can be considered a very basic design problem as it involves accepting a requirement ("this screw must be unscrewed"), knowing that screwdrivers can be fabricated from the materials at hand, and then using one's understanding of the domain to develop a workable solution. In NARS' case, it uses ampliative reasoning to solve this problem in a few steps of non-axiomatic reasoning [17, 24].

In AERA's case this was addressed using AERA's mechanism of causal-relational models and composite states [17]. Given only basic causal models ('lighters can produce fire,' 'fire cause things to heat up,' 'heated plastic becomes pliant,' 'pliant substances can be deformed,' and 'waiting can allow things to cool and harden'), AERA discovered the same plan as NARS and was able to, in a very simple simulated environment, successfully unscrew the screw. In **Fig. 1**, we can observe the first step of AERA's planning starting on the right-hand side of the diagram with the goal that screw 's' must have state 'unscrewed.' It should be noted that the knowledge AERA uses to perform the task does not come from human hand-coding; rather, the names have been added for readability. AERA then notes that this is possible if it instantiates its 'UNSCREW' model, and that in order to do so it must first have some toothbrush 't' with the 'cooled' state. Briefly summarizing, a similar reasoning process then continues to work backwards linking a 'WAIT' model that will allow the toothbrush to cool, a 'FORM' model that will shape it into a screwdriver shape, and a 'HEAT' model that will heat it up in the first place.[7] With a complete plan formulated, AERA can then execute its solution to fulfill the design requirement of unscrewing the screw.

This simplified example demonstrates the principles and potential applications of autonomous causal reasoning in conceptual design; when given sufficient information and the ability to investigate and make its own plans, a GMI-aspiring agent was able to design a solution to a problem.

6 Conclusions & Future Work

We posit that research on general machine intelligence, unlike other AI research, is necessary to address the challenge of synthetic engineering agents capable of conceptual design. The five requirements we have identified support such a stance, as few if any other fields of research are better suited to address them in the necessary unified manner. Understanding the causal nature of reality and how that influences the problem-solving process, creatively contributing to impasses in an explainable manner (where multiple options are available), and autonomously decomposing problem requirements in order to make design decisions are all central faculties for any engineer, human or otherwise

While existing architectures such as AERA and NARS have only recently been tested in this application, they should already meet the requirements to perform these tasks. The next step in this work will be a more advanced test implementation of a design agent. This work will most likely apply AERA, which already presents important functionalities including autonomous causal modeling and unified abductive and deductive non-axiomatic reasoning; we believe this will be of help in realizing a first implementation. However, as stated in the introduction, it is quite possible that the listed requirements, while necessary, are not sufficient to realize artificial agents capable of autonomous conceptual engineering design. For now, however, they provide a viable next step in a search for an answer.

Disclosure of Interests. The authors have no competing interests to declare.

[7] Refer to Section 6.2 of [17] for the full analysis.

References

1. AAAI 2025 Presidential Panel on the future of AI research (March 2025), AAAI Press
2. Altavilla, S., Blanco, E.: Are AI Tools Going to be the New Designers? A Taxonomy for Measuring the Level of Automation of Design Activities. Proceedings of the Design Society: DESIGN Conference **1**, 81–90 (2020), publisher: Cambridge University Press
3. Belenchia, M.: Towards a Theory of Causally Grounded Tasks. Ph.D. thesis, Reykjavík University (2021), https://skemman.is/bitstream/1946/39956/1/msthesis.pdf
4. Bhatt, A.N., Majumder, A., Chakrabarti, A.: Analyzing the modes of reasoning in design using the SAPPhIRE model of causality and the Extended Integrated Model of Designing. AI EDAM **35**(4), 384–403 (2021)
5. Campbell, M.I., Cagan, J., Kotovsky, K.: A-Design: Theory and Implementation of an Adaptive, Agent-Based Method of Conceptual Design. In: Gero, J.S., Sudweeks, F. (eds.) Artificial Intelligence in Design '98, pp. 579–598. Springer Netherlands, Dordrecht (1998)
6. Castro Pena, M.L., Carballal, A., Rodríguez-Fernández, N., Santos, I., Romero, J.: Artificial intelligence applied to conceptual design. A review of its use in architecture. Automation in Construction **124**, 30 (2021)
7. Department of Computer and Information Sciences, Temple University: OpenNARS for Applications (2020), original-date: 2020-02-16T17:33:05Z
8. Eberding, L., Thompson, J., Thórisson, K.R.: Argument-driven planning & autonomous explanation generation. In: Proc.Artif. General Intell. (2024)
9. Ege, D.N., Øvrebø, H.H., Stubberud, V., Berg, M.F., Elverum, C., Steinert, M., Vestad, H.: ChatGPT as an inventor: eliciting the strengths and weaknesses of current large language models against humans in engineering design. AI EDAM **39**, 15 (2025)
10. Emdanat, S., Stiny, G., Vakaló, E.G.: Generative Systems in Design. AI EDAM **13**(4), 239–240 (1999), publisher: Cambridge University Press
11. Gu, E.H., Chandrasegaran, S., Lloyd, P.: Synthetic users: insights from designers' interactions with persona-based chatbots. AI EDAM **39**, 17 (2025)
12. Ma, P., Wang, T.H., Guo, M., Sun, Z., Tenenbaum, J.B., Rus, D., Gan, C., Matusik, W.: LLM and Simulation as Bilevel Optimizers: A New Paradigm to Advance Physical Scientific Discovery (2024). https://doi.org/10.48550/arXiv.2405.09783, http://arxiv.org/abs/2405.09783, arXiv:2405.09783 [cs]
13. Nivel, E., Thórisson, K.R., Steunebrink, B.R., Dindo, H., et al.: Bounded Recursive Self-Improvement. Tech. rep., Dept. Comp. Sci., Reykjavik U. (2013)
14. Nordlund, M., Lee, T., Kim, S.G.: Axiomatic Design: 30 Years After. In: Volume 15: Advances in Multidisciplinary Engineering. p. V015T19A009. American Society of Mechanical Engineers, Houston, Texas, USA (2015)
15. Pearl, J.: Bayesianism and Causality, or, Why I am Only a Half-Bayesian. In: Gabbay, D.M., Barwise, J., Corfield, D., Williamson, J. (eds.) Foundations of Bayesianism, vol. 24, pp. 19–36. Springer Netherlands, Dordrecht (2001)
16. Royzen, Z.: Application TRIZ in Value Management and Quality Improvement. In: SAVE Annual Proceedings. vol. XXVIII, pp. 94–101 (1993)
17. Schaff, C.: A Foundation for Autonomous Conceptual Engineering Design. M.Sc. thesis, Department of Computer Science, Reykjavík University, Iceland (2024)

18. Sheikhlar, A., Thorisson, K.R., Thompson, J.: Explicit General Analogy for Autonomous Transversal Learning. In: Proceedings of Machine Learning Research. vol. 192, pp. 48–62 (2022)
19. Smithers, T.: Design as Exploration: Puzzle-Making and Puzzle-Solving. In: Exploration-based models of design and search-based models of design. Carnegie Mellon University, Pittsburgh (1992)
20. Steunebrink, B.R., Thórisson, K.R., Schmidhuber, J.: Growing Recursive Self-Improvers. In: Steunebrink, B., Wang, P., Goertzel, B. (eds.) AGI 2016. vol. 9782, pp. 129–139. Springer International Publishing, New York, New York (2016). https://doi.org/10.1007/978-3-319-41649-6_13
21. Thórisson, K.R., Kremelberg, D., Steunebrink, B.R., Nivel, E.: About understanding. In: Proc. Artificial General Intelligence. pp. 106–117 (2016)
22. Thórisson, K.R., Helgason, H.P.: Cognitive architectures & autonomy: A comparative review. Journal of Artificial General Intelligence **3**, 1–30 (2012)
23. Thórisson, K.R., Rörbeck, H., Thompson, J., Latapie, H.: Explicit goal-driven autonomous self-explanation generation. In: Proc. Intl. Conf. Artif. General Intell. pp. 286–295 (2023)
24. Thórisson, K.R., Schaff, C.A.: High-level conceptual design automation requires ampliative reasoning. In: Proc. NordDesign'24 (2024)
25. Thórisson, K.R., Talbot, A.: Cumulative learning with causal-relational models. In: Proc. Artificial General Intelligence (2018)
26. Vidner, O., Wehlin, C., Persson, J.A., Ölvander, J.: Configuring Customized Products with Design Optimization and Value-Driven Design. Proceedings of the Design Society **1**, 741–750 (2021), publisher: Cambridge University Press
27. Wang, P.: Rigid Flexibility: The Logic of Intelligence. Springer Science (2006)
28. Wang, P.: Case-by-Case Problem Solving. In: Proceedings of the 2nd Conference on Artificial General Intelligence (2009). Atlantis Press, Arlington, Virginia, USA (2009). https://doi.org/https://doi.org/10.2991/agi.2009.43
29. Wang, P.: Fundamental Limitations of Large Language Models in Reasoning (Apr 2025), https://www.youtube.com/watch?v=PXqnqmm_UfI
30. Žavbi, R., Duhovnik, J.: Conceptual design of technical systems using functions and physical laws. AI EDAM **14**(1), 69–83 (2000), publisher: Cambridge University Press.