NORDDESIGN CONFERENCE
REYKJAVIK, ICELAND, 12-14 AUGUST 2024

High-Level Conceptual Design Automation Requires Ampliative Reasoning

Kristinn R. Thórisson[1,2] & Chloe Shaff [1]

[1] Center for Analysis & Design of Intelligent Agents, Reykjavik University
[2] Icelandic Institute for Intelligent Machines, Reykjavik, Iceland

Abstract: Engineering design is a problem-solving process that works from a high-level description of a problem or plan and proceeds to iteratively define an increasingly detailed solution aimed to meet the criteria required for physical implementation. While significant efforts have been made to automate lower- and mid-level design tasks, much less work has been done on the automation of high-level conceptual design. In the pursuit of novel solutions, reliable engineering design must inevitably proceed through an iterative analysis-refinement process. To be effective and efficient for novel and reasonably complex design tasks, spanning several levels of detail, these iterations cannot be done through a random search, as it would be much too slow for practical use. Instead, it must be guided by knowledge, where each iteration proceeds through verifiable arguments about why the current solution is an improvement over what came before; a process which requires reasoning across multiple modalities and levels of detail. Addressing the full scope of potential solutions, the types of reasoning required for producing an acceptable solution cannot be prescribed or be known beforehand, any more than the solution itself. Verifiable arguments, in turn, must be based on good models of cause-effect relations. In other words, high-level design *must rely on ampliative reasoning over knowledge of cause-effect relations*. Unlike some low-level design, many contemporary methods in artificial intelligence therefore fall short for reliable automation of high-level design tasks. We present arguments for these conclusions, explore the nature of the high-level design process, and discuss frameworks for automating it.

Keywords: Conceptual design, artificial intelligence, causal reasoning, automation, explanation

1 Introduction

Every design starts with a problem to be solved. This often comes in the form of some design task with a given set of constraints that a designer seeks to satisfy. The design process thus consists in progressing from that initial specification to an acceptable solution. Since even redesign involves modifying or extending existing solutions, design tasks always involve some level of novelty. Here, we are most interested in the end of the spectrum containing greater novelty. Working from this position, engineering design can be seen to consist of two qualitatively distinct design regimes that we can refer to as high-level (HL) and low-level (LL) design (Schaff, 2024). LL design is simpler (though by no means simple) and defined as the *specification of the concrete geometry and interaction of parts*, while HL design is more abstract, defined as the *conceptual problem solving process leading* a designer incrementally from a problem description *to increasingly better intermediate solutions* (a 'better' solution includes more detail and is closer to a complete specification). For example, an HL design of a briefcase might discover a solution with a handle and a lockable container. From here, the LL design can determine how exactly the handle should be shaped, how big the container should be, and how it is all expected to be manufactured. One difference between these two stages consists of the size of the solution space being addressed: LL part design starts with numerous constraints all at the same level of detail while HL conceptual design starts with abstract problem descriptions with specifications at varying levels of detail and a non-finite (or near infinite) number of possible solutions. LL design calls for knowledge and expertise in particular domains such as materials, electronics, machinery, etc.; HL design relies additionally on more general knowledge (subsuming to some extent lower levels).

Here we focus on HL engineering design in domains calling for reliability, safety and transparency, like air traffic control, medical device design, and safety-critical infrastructure, where reliability can be backed up with arguments and valid explanations for why the solutions should be trusted. Systems that can do so automatically, over knowledge they have learned autonomously, are few and far between; here we look briefly at two such systems, the Autocatalytic Endogenous Reflective Architecture (AERA; Nivel et al. 2012) and Non-Axiomatic Reasoning System (NARS; Wang, 2006).

The value of developing a system capable of reliable, well-argued autonomous conceptual engineering design (ACED) is similar to the value of any kind of trustworthy automation; whether to improve safety, increase speed, reduce cost or risk, the outcome can typically be translated directly to monetary value. A computer system capable of a variety of ACED tasks could presumably increase the output of existing engineering teams substantially. Such a system could even be made to produce solutions too complex or involved for a conventional human-centric design process. ACED agents

could even have the ability to take on design processes largely on their own, acting with a high degree of agency. In order to fully take advantage of this capability, however, the system must be trustworthy. Human engineers must be able to inspect its reasoning, evaluate its proposed solutions, and validate its work. This requires explanation and a certain amount of reasoning ability.

Considering the potential advantages of the technology, substantial investigation has already gone into the process of autonomous and automated engineering design (Venkataraman & Chakrabarti, 2010, Bhatt et al., 2021, Kügler et al., 2023). Attempts have been made to formalize the design process and represent it in terms of activities, design outcomes, and the parameters that lead to a strong design. Additionally, there is already an extensive body of work in low- and mid-level design automation in terms of generative design[1] (GD) and knowledge-based engineering (KBE; Kügler et al., 2023), respectively. While powerful, these methods are not compatible with the nature of the larger HL design space, which requires a more high-level exploratory approach. Finally, work in task theory (Belenchia et al., 2021) can aid in the formal representation and analysis of engineering design problems. All of this is to say that, while comparatively little work has been focused on the creation of a true ACED-capable agent, there is a significant body of work on which to build such research.

Given the iterative nature of the design process and its reliance on questioning select assumptions and exploring alternatives, we posit that any ACED-capable agent must be capable of a unified abduction-deduction process. Encapsulating this process is the concept of ampliative reasoning, which combines (non-axiomatic, defeasible) deduction, abduction, induction and analogy (cf. Sheikhlar et al. 2021).[2] This is to say that an agent capable of ampliative design can build models from observation, update and correct them over time, use them to explain phenomena, use them to plan actions, and compare them to other similar models, in an effort to iteratively improve its design solution. All of these capabilities would enable a design agent to fully understand a problem, work problems at multiple levels of detail, and do all of this with transparent reasoning that could be explained to or inspected by human engineers.

In this paper we detail how an ACED must be based on ampliative reasoning, combining causal knowledge and argumentation to reason about the problem at hand, at multiple levels of detail, based on verifiable and explainable arguments for its design choices, and discuss cognitive architectures that support such an approach.

2 Related Work

Great many attempts have been made to systematize and automate engineering design. In the GEMS model (Venkataraman & Chakrabarti, 2010), design is considered as a series of the discrete activities: *generate*, *evaluate*, *modify*, and *select;* new ideas are generated, their potential evaluated, changes made in light of feedback, and the process is repeated until a final design is selected. An alternative approach proposed by a related research team, the SAPPhIRE model (Bhatt et al., 2021), which considers designs in terms of how design outcomes change over time, through a progression of ideas from the high-level reasoning about function to low-level selection and design of individual parts. This latter approach is particularly powerful when the domain is well-known, as the agent can work its way through existing knowledge from a desired function to the processes that perform that function and then on to the physical affectations and implementation of the function into a part. The challenge with this approach is that an engineer must also be able to work with uncertainty, which is not addressed directly in these approaches.

Generative design (GD) is the practice of using optimization processes such as genetic algorithms to explore a solution space and solve a low-level (LL) design problem for a particular set of constraints, loads, and other features. It is now a well-developed technology and is even available to the general public. GD excels in creating optimized and unique designs for parts even when the conditions are particularly challenging or novel. However, it can only work at this lowest level of detail; GD is based on the direct manipulation of geometry and a larger-scale or higher-level design problem would result in a significant expansion in search space, not to mention the difficulty of developing an optimization process capable of this greater challenge. One possible solution is knowledge-based engineering (KBE), which encodes the knowledge of human engineers into conceptual frameworks that can speed the mid-level design process and increase the extent to which institutional knowledge is preserved and reused (Kügler et al., 2023). These systems can function much like templates for complex processes in which a specific set of design specifications is entered at the beginning and a finished design is returned at the end. This makes them quite suitable for mass customized products; KBE systems are even beginning to be offered to the general public.[3] The difficulty with KBE systems is that

[1] See e.g. *Autodesk, 2023: What is Generative Design | Tools | Software;* https://www.autodesk.com/solutions/generative-design – *accessed Oct. 4, 2023*.

[2] Note that this differs from the definition that some authors use, wherein 'ampliative reasoning' refers to methods relying on abduction and induction in some combination (cf. Psillos, 2011).

[3] See e.g. *ParaPy—Knowledge-Based Engineering Platform [WWW Document]*. https://parapy.nl/ – *accessed Oct. 25, 2023.*

their knowledge is static as most of them involve no learning; they are intended to be highly reliable so they are built and tested by hand.

Smithers (1992) suggests that design is not just a simple search process but an active effort to explore and experiment with possible solutions. It is rare that a problem is perfectly described at the outset of a design exercise and so, as a solution is sought, an engineer should expect to discover imperfections in the problem description. They should be able to work to resolve these imperfections either by communicating with their client and changing their design or by finding a solution on their own. Then, there is the question of the exact way in which designs are simplified and guided through the conceptual stage. Schut (2010) takes the approach of removing degrees of freedom to keep the solution space minimal, while Axiomatic Design (Nordlund et al., 2015) tries to minimize information content by enforcing modularity and controlling dependencies in the problem description itself. The difficulty is in implanting this into a software agent as this will require systems with the agency to learn on their own, experiment with solutions, and update their knowledge as best practices change. Even Axiomatic Design's translation of customer needs to functional requirements involves some level of understanding of the problem itself; it simply cannot be done with low-level optimization processes that work below this level. Perhaps a hybrid KBE and GD agent could do this by reasoning through a solution and then dispatching a generative designer to finish part designs. This, however, will still require a powerful cognitive architecture to support the integration of these two approaches and to provide the learning capabilities desired for an ACED agent.

Finally, the issue of problem analysis has been subjected to some scrutiny in work on task theory (Thórisson et al., 2016a), a budding field aimed at creating a theoretical basis for intelligent task execution at a fundamental level. Task-theoretic methods represent tasks as networks of causal-relational models (CRMs) that can be used to describe knowledge of them (cf. Belenchia et al., 2021; Eberding et al., 2021). By chaining such models forward (deduction) and backward (abduction), an agent can explain its conceptualization of phenomena and develop plans for achieving goals. One of the many interesting features of task theory is how a task's level of detail is captured, as changes in the level of detail can significantly change the task itself (Belenchia et al., 2021:25). In this view, tasks can be constructed as a hierarchy with respect to an agent's goals. Breaking tasks down and analyzing them this way makes solution generation significantly more straightforward. This more rigorous approach also lends itself well to the development of an ACED agent. Below we show how these ideas can be brought to bear on the problem.

3 The Engineering Design Process: Novelty & Iteration

Most design tasks are vague and significantly under-constrained and most design tasks in the physical world involve several levels of detail. As soon as a solution for a particular design problem exists, there is no need for engineering design – engineering design thus always involves novelty. Smithers (1992) considers engineering design as not just a progression towards a (new) solution but also to a complete and unambiguous problem description, proposing the idea of an Initial Requirement Description (IRD), which "may be incomplete, inconsistent, imprecise, and ambiguous or (more typically) some combination of all of these"; there is the potential for missing or contradictory criteria, criteria that subsume other criteria, and so on. Reconciling these criteria is among the first steps of the design process, clearly separating high-level (HL) and low-level (LL) design. This requires further specification and analysis which, in turn, requires reasoning about the problem at a conceptual level. The designer must grasp the intent and prerequisites behind each criterion and all of the relationships that might be present between the criteria. The process as described involves doing a breadth-first exploration of the design space, calling for knowledge-based interpolation and extrapolation to fill in missing constraints, which can only be done through (defeasible, non-axiomatic) induction (cf. Sheikhlar et al. 2021).

All tasks come with acceptable tolerances, T, for the solution's form and function. A *design task*[4] involves producing a solution x that meets these requirements, decreasing the difference, $\Delta(x)$, between the specification and the solution, as the task progresses. As the design gets closer to an acceptable finished solution (S), the detail that is being addressed necessarily increases;

$$\lim_{x \to S} \Delta(x) \leq T \quad . \qquad (1)$$

This relationship also holds for how levels of detail (LoD) can be efficiently addressed during the design process: The HL engineering design process involves working through problems at multiple levels of detail, drawing on knowledge from a variety of sources, and refining solutions iteratively until a problem is declared 'solved.' For any reasonably complex design task, e.g. the design of a new aircraft, such iteration could not be based on random search, as this would extend the design time to impractical levels. When starting out a design, based on a complete specification, questions about low-level solutions may still need to be explored. Exploring every potentially-relevant small detail for every

[4] In our terminology, a *design* is a blueprint for a solution to a role or function; a *design problem* is a call for a design solution; a *task* is an assigned problem that may include additional requirements (e.g. methods); a *design task* is thus an assigned design problem.

high-level design option is not possible, however, due to time, space and energy constraints. So in HL design, any finer-grain LoD that might need to be explored must be considered for being sufficiently relevant or not – going into more details during high-level design will slow down the design process. If *all* details are relevant, the high-level design cannot proceed until *all* questions about the details have been answered; in this case the time for completing the coarse-grain design is equal to completing the whole design including *every* detail (a worst-case scenario with respect to design cost). Of course, deciding which levels of detail matter during the initial design space exploration, and should be investigated more deeply, must depend on verifiable arguments, which in turn must rely on valid cause-effect relations. So random exploration is clearly also not an option for an effective iterative design process. (A worst-case thought experiment makes this clear. Imagine employing random search to design an airliner: With a volume of 80×80×25 meters, a spatial resolution of 0.1 mm, 15 different material options for each voxel, and a generation rate of 1 billion design variations per second, this task would take on the order of $10^{18,800}$ years, not counting the testing of the outcome).

This introduces an additional challenge in HL design, in that HL models of the world must be based on observations of LL processes. Forming and using these models requires an agent to move between different levels of detail; deciding whether some design option is more worthy of exploration than another must involve, at the very least, abduction, which is to say, arguments for what could be prerequisites for particular design outcomes. But most of the time it would also involve deductions, especially when investigating implications of specific design choices, e.g. the value of some important variable, like the size of an automobile's gas tank or the wingspan of an airplane. The net outcome of this is that unified and iterated deduction–abduction cannot be avoided for HL design. This holds whether the reasoning is done to produce arguments for a particular path or to prepare for doing an actual experiment in the world, to test a specific assumption which would then be integrated back into the HL plan.

Proposing (and separating) solutions that may work, over myriads of possible designs that don't work, is an important part of high-level design space exploration. In highly non-linear design domains such as the physical world, minute variations in any proposed solution may instantly reduce it to a non-solution. The only practical way to consistently and reliably evaluate alternatives in such domains is through reasoning over alternatives in light of known cause-effect relationships (Thórisson & Talbot, 2018; Pearl, 2009; Halpern & Pearl, 2005) between the nature of the problem and the targeted solution's purpose, design constraints, and context. Human designers use their knowledge of how the world works, coupled with argumentation, to reason through alternatives and rule out bad ideas, paths, and options. Doing so at a high-level of design (coarse-grained level of detail) up front can shave off enormous amounts of unnecessary design time. But to do so, they must be grounded (i.e. the knowledge must model the target phenomena's useful cause-effect relations; cf. Belenchia et al. 2021; Eberding 2021).

An important part of this process is the decomposition of large problems into sub-problems. Returning to the briefcase example from the introduction, it is not enough to know that the solution needs a locked container on a handle: we also need to know how big the container is, what the handle should be made of, whether there are any aesthetic preferences, and so on. Exploring these sub-problems requires either a pre-existing knowledge of the problems at hand or the ability to make analogies with similar problems whose solutions are known. In the analogy sense, we can see how a design agent could see similarities between its current problem and solutions to problems it has worked in the past. For instance, a designer might have prior experience with briefcases and know that they often contain paper somewhere around the A4 and Letter sizes. They could then adapt this solution to the current problem through modification, to speed up the process (cf. Sheikhlar et al. 2022). Without the ability to compare and contrast knowledge, our designer might have had to solve the problem from scratch, losing valuable time and resources. Analogy is also important for supporting generalization (cf. Sheikhlar et al. 2021).

Taken together, the above argues for abduction and deduction when considering which levels of detail must be explored further during HL design, analogy for repurposing and reuse of knowledge, and induction for creating rules of thumb and generalizing. The order of these, however, will be difficult to systematize, as the web of dependencies between the sub-components of the potential solution and the order in which it is being explored may steer the design process in various directions. The order of applying different kinds of reasoning to answer the questions as they arise must be flexible. Like a human designer, an ACED agent will thus need to be able to take a hybrid approach by using prior knowledge and a variety of reasoning methods to guide its solution creation, and even be able to fall back to blind experimentation when informed methods fail completely (cf. Thórisson 2020). The picture that emerges is clearly one of an ACED agent capable of ampliative reasoning.

In summary, engineering design is a process highly dependent on the agent's ability to conceptualize the problem and the solution. They must be able to analyze the problem requirements and understand how they relate to the context of the problem. They must be able to propose and evaluate different solutions and explain the reasoning for their decisions. Finally, they must be able to bridge gaps in their understanding by developing context-dependent rules of thumb from their observations of the world. All of this must be done at the conceptual level and must be represented with causal

reasoning, there is simply no way to perform this kind of conceptual problem-solving without ampliative reasoning and understanding.

4 Explanation is Integral to Engineering Design

From the above analysis it becomes clear that explanation must play a role in the choices that an engineering design agent performs. In the history of AI, the concept of 'explanation' was for the longest time merely a footnote. It has recently increased in importance and is now considered by many a central concept in the pursuit of general intelligence (cf. Thórisson et al., 2023; Thórisson & Minsky, 2022). This heightened focus is primarily linked to the contemporary surge in the use of AI systems designed for the automation of diverse industrial functions (cf. Weimer, 2016), involving a variety of applications such as insurance risk assessment, industrial inspection, job hiring, language translation and visually-guided control of automobiles. With features unmatched by other technologies, artificial neural networks (ANNs) have emerged as a pivotal technology for these purposes. With appropriate training, ANNs can seemingly turn extensive datasets into useful knowledge through a comprehensive network of classification functions (Wang & Li, 2016). Their use for practical situations, where plenty of data is already available, makes their deployment in many cases far less costly than other alternative approaches including manual coding.

However, the motivation for pursuing explanations in large ANN systems does not arise primarily from a specific desire for explanation capabilities; instead, it originates from the ANNs' inherent opacity and the requirement that automation be trustworthy (cf. Wing, 2021). Trustworthy automation requires, among other things, that a system's operation be explainable, especially in light of errors. Like any human-made technology, ANNs are not infallible. However, elucidating with precision and coherence, at varying levels of detail and abstraction, the process by which any deployed ANN produces its output, has proven to be a formidable challenge. This difficulty may, to a considerable extent, be attributed to the technology's inability to represent causal relations explicitly. The task of explaining their operation cannot be relegated to the ANN systems themselves because they do not have the capacity to reliably reflect on their own operation, which stems from their inability to (a) learn as they go and (b) represent cause-effect relations. Presently, these systems are not suitably positioned for routine explanations in practical contexts. For any safety-critical engineering design, and in fact most designs in human society, no obvious solution is in sight for putting ANN-based technologies at the center of such operations.

A sound explanation effectively addresses blind spots, provides clarity, and highlights previously obscure elements. Crucially, a good explanation adheres to implicit (or explicit) constraints, avoiding any breach of relevant rules. Mere reference to correlational data is insufficient for achieving this; the foundation must instead rest on authentic and relevant causal relations (Thórisson et al. 2023). A sound explanation's essence lies in illuminating or highlighting why a given phenomenon – whether it involves a sequence of events, a particular situation, or any other outcome – takes a specific form, rather than an alternative one (Pearl, 2009; Halpern & Pearl, 2005).

Thórisson et al. (2023) characterize explanation as a goal-driven process and argue that the quality of an explanation generation mechanism is based on how well it fulfills three purposes – or goals: uncovering unknown or hidden patterns, highlighting or identifying relevant causal chains, and identifying incorrect background assumptions. At the heart of any good explanation lies knowledge of cause-effect relations. Halpern and Pearl (2005:851) state: "…the role of explanation is to provide the information needed to establish causation. Thus, as we said in the introduction, what counts as an explanation depends on what one already knows." The process of engineering design involves what Thórisson et al. call 'self-explanation,' that is, the repeated explanation generation to oneself for the purpose of uncovering inconsistencies in one's own knowledge.

For reasons involving safety, efficiency, and trustworthiness, we argue that explanation is critical in systems performing HL design tasks. Existing architectures such as ANNs are powerful but their opacity makes it difficult to fully trust — and therefore utilize — them in a design workflow. The solution to this is a system not just capable of explanation but based on it. A cognitive architecture capable of self-explanation would be a strong foundation on which to build an ACED-capable agent.

5 Cognitive Architectures for the Engineering Method

As we have detailed above, any engineering design process will inevitably involve iterative refinement of requirements to an increasingly smaller set of options, specifications, choices, and solutions. An autonomous conceptual design agent (ACED) should be able to refine its understanding[5]of a problem as it works on it, striving to develop increasingly better – and unified – models of the subject matter and the context in which a solution shall operate. Since the agent will be performing thought experiments as well as physical experiments on the world (especially where knowledge and

[5] We follow the definition of 'understanding' provided by Thórisson et al. (Thórisson 2016b; Bieger & Thórisson, 2017), which proposes an information theory of the process of understanding compatible with the concept of 'sense-making' (Klein et al. 2007).

imagination falls short) and weaving this into a cohesive solution, we can liken this to a process of 'argument-sustained exploration.' Since information is very often missing in the early stages of engineering design, any practical machine capable of autonomous high-level (HL) design must be able to learn as it goes (cf. Thórisson et al., 2019) – it cannot assume, for every novel design it undertakes, that it already knows what is necessary for an acceptable solution. We should also remind ourselves that most if not all design is novel, at least to the designer, as otherwise it would not require a new design.

With this in mind, the limitations in one potential methodology that is often mentioned in the context of design, artificial neural networks (ANNs). While ANNs are a powerful technology for solving many well-defined problems for which plenty of data is already available, they do not permit the kind of reasoning with causality and iterative reflection that is needed for developing the increasingly better understanding of a problem required in the engineering design process (Arkoudas, 2023). These shortcomings have been to some extent addressed in KBE systems (Kügler et al., 2023), and while they represent an improvement in reasoning capacity over ANNs, a KBE-based approach will likely not be able to experiment and develop its knowledge of the world over time, and will not learn. They will also not be well-suited to generate verifiable explanations for their design choices. Thus, neither neural nor purely knowledge-based architectures are well-suited to address (novel) design problems that require safety-critical circumstances.

Any artificial agent capable of engineering design must be able to transform functional requirements into parts, explore in less-than-perfectly understood environments, and take the agency to keep solutions simple and abstract for efficiency. This is what the general intelligence of human design engineers achieves, and while no generally intelligent systems have been created to date, some implemented cognitive architectures currently under development show promise in this regard. The NARS[6] architecture, for instance, has been demonstrated to find design solutions to simple challenges, such as refashioning a toothbrush into the shape of a screwdriver when faced with the task of loosening a screw without having a screwdriver[7]. The system builds on non-axiomatic reasoning, which assumes that any statement learned about the world is defeasible (Pollock, 2010), when conflicting evidence arises.

Another system that could be considered to address the requirements of reliable automation of high-level engineering design is the Autocatalytic Endogenous Reflective Architecture[8] (AERA; Nivel et al., 2013; Thórisson 2020). Developed through a methodology that has embraced the requirements of exploration and self-reflection, the Constructivist AI (Thórisson, 2012), an AERA agent, instead of being limited to its initial knowledge or the knowledge provided by its creators at 'birth,' can autonomously grow its knowledge from a small initial seed, and thus be well-suited to an environment of exploration and learning (Thórisson 2020).

In terms of engineering methodologies that may be tested in AERA, of particular interest is the approach taken by Žavbi & Duhovnik (2000), who propose that solving engineering problems involves chaining together equations of physical phenomena to describe a device's transformations of energy and forces. Their approach has been successfully used to reinvent, at a conceptual level, devices as complex as microphones. While they do not propose how such a system could automatically create these equations from its experience and observations of the world, this is solved in the AERA system, which can learn about tasks cumulatively, representing knowledge in terms of causal-relational models (CRMs; Thórisson & Talbot, 2018). CRMs are simple enough that they can be created and updated autonomously by the system itself; they can be chained together to generate explanations of observed phenomena, produce plans to achieve goals, and generate predictions in real time. The challenge of automating engineering design could possibly be addressed by including a CRM-based approach to problem description and using AERA or NARS to implement the high-level reasoning at the core of the process. Indeed, this has been the subject of a recent MSc thesis by one of the authors of this paper (Schaff, 2024).

In order to achieve a system capable of ACED, we need a new cognitive architecture capable of learning, based on ampliative reasoning over cause-effect knowledge. NARS and AERA show how this may be quite possible and feasible, presenting research platforms and open-source software for future development. Due to these systems' autonomous cumulative learning and built-in ampliative reasoning, they are among the very few implemented cognitive architectures to be suited for research in reliable high-level design automation.

6 Conclusions & Future Work

Due to the nature of high-level design tasks, agents capable of autonomous conceptual engineering design (ACED) will likely not resemble those that have been developed for low- and mid-level tasks. An artificial system capable of conceptual engineering design, to be fully autonomous, must have abilities to reason about problem requirements, explore novel solution spaces, iterate over the problem and solution, work across differing levels of detail, reuse parts of

[6] Non-Axiomatic Reasoning System; http://www.opennars.org (cf. Wang, 2006) – *accessed Feb. 13, 2024*.
[7] Available at https://github.com/opennars/OpenNARS-for-Applications/blob/master/examples/nal/toothbrush.nal
[8] Autocatalytic Endogenous Reflective Architecture (cf. Nivel et al., 2013); see http://www.openaera.org – *accessed Feb. 13, 2024*.

existing solutions to speed the design process, and explain their reasoning to human engineers. To date, no technical solution exists that can address this other than non-axiomatic defeasible ampliative reasoning. Existing systems using genetic algorithms, artificial neural networks, and knowledge-based architectures, while useful for many things, are unable to operate at the level of conceptual reasoning demonstrated by humans. For this reason we suggest that future research into high-level conceptual design automation focus on neuro-symbolic (cf. Latapie, 2022) or neo-symbolic foundations such as demonstrated in AERA (Nivel et al. 2013) and NARS (Wang, 2006). The work presented tells us that the process of engineering design is very close to the heart of general intelligence. While work on creating machines with general intelligence is a formidable undertaking, an autonomous conceptual engineering design agent might, however, be somewhat closer at hand if we build on already existing research aiming for this goal.

Acknowledgments

The authors would like to thank the General Machine Intelligence groups at CADIA, Reykjavik University, and the Icelandic Institute for Intelligent Machines, for numerous discussions on the topics covered in this paper.

References

Arkoudas, K. 2023. GPT-4 Can't Reason. https://arxiv.org/abs/2308.03762 – *accessed May 1st, 2024*.

Belenchia, M., K. R. Thórisson, L. M. Eberding & A. Sheikhlar, 2021. Elements of Task Theory. Proc. Intl. Conf. Artif. General Intell. (AGI-21), 19-29.

Bhatt, A.N., Majumder, A., Chakrabarti, A., 2021. Analyzing the modes of reasoning in design using the SAPPhIRE model of causality and the Extended Integrated Model of Designing. AI EDAM 35, 384–403. https://doi.org/10.1017/S0890060421000214 - *accessed Feb. 13, 2024*.

Bieger, J. & K. R. Thórisson, 2017. Evaluating Understanding. IJCAI-17 Workshop on Evaluating General-Purpose Intelligence, International Joint Conference on Artificial Intelligence, August 20, Melbourne, Australia.

Eberding, L. M., M. Belenchia, A. Sheikhlar and K. R. Thórisson, 2021. About the Intricacy of Tasks. Proc. Intl. Conf. Artif. General Intell. (AGI-21), 65-74.

Halpern, J.Y., Pearl, J., 2005. Causes and explanations: A structural-model approach — Part II: Explanations. Brit. J. Phil. Sci. 56, 843–847.

Klein, G., J. K. Phillips, E. L. Rall & D. A. Peluso, 2007. A Data–Frame Theory of Sensemaking. New York: Taylor & Francis Psychology Press.

Kügler, P., Dworschak, F., Schleich, B., Wartzack, S., 2023. The evolution of knowledge-based engineering from a design research perspective: Literature review 2012–2021. Adv. Eng. Inform. 55, 101892. https://doi.org/10.1016/j.aei.2023.101892 - accessed Feb. 13, 2024.

Latapie, H., Kilic, O., Thórisson, K. R., Wang, P. & Hammer, P. 2022. Neurosymbolic Systems of Perception and Cognition: The Role of Attention. *Front. Psychol., 20 May, Sec. Cognitive Science*.

Nivel, E., Thórisson, K.R., Steunebrink, B.R., Dindo, H., Pezzulo, G., Rodriguez, M., Hernandez, C., Ognibene, D., Schmidhuber, J., Sanz, R., Helgason, H.P., Chella, A., Jonsson, G.K., 2013. Bounded Recursive Self-Improvement. Reykjavik University School of Computer Science Technical Report, RUTR-SCS13006 / arXiv:1312.6764 [cs.AI] http://arxiv.org/pdf/1312.6764v1 – *accessed Feb. 13, 2024.*

Nordlund, M., Lee, T., Kim, S.-G., 2015. Axiomatic Design: 30 Years After. Proc. International Mechanical Engineering Congress and Exposition (ASME 2015), vol. 15, paper num. IMECE2015-52893, V015T19A009. Houston, TX: American Society of Mechanical Engineers, Advances in Multidisciplinary Engineering. https://doi.org/10.1115/IMECE2015-52893

Pearl, J., 2009. Causality: Models, Reasoning and Inference. Cambridge University Press, New York, NY, USA, 2nd edn.

Pollock, J. L., 2010. Defeasible reasoning and degrees of justification. Argument & Computation, **1**(1):7–22.

Psillos, S., 2011. An explorer upon untrodden ground: Peirce on abduction. In: Handbook of the History of Logic, 10:117–151. Elsevier.

Schaff, C. 2024. A Foundation for Autonomous Conceptual Engineering Design. Master of Science thesis, Department of Computer Science, Reykjavik University.

Sheikhlar, A., Eberding, L.M. & Thórisson, K.R., 2021. Causal Generalization in Autonomous Learning Controllers. Proc. Artificial General Intelligence (AGI-21), 228-238.

Sheikhlar, A., Thórisson, K. R. & Thompson, J., 2022. Explicit General Analogy for Autonomous Transversal Learning. Proc. Machine Learning Research, **192**:48-62.

Schut, E.J., 2010. Conceptual Design Automation: Abstraction complexity reduction by reasilisation and knowledge engineering. PhD Thesis, Technische Universiteit Delft.

Smithers, T., 1992. Design as Exploration: Puzzle-Making and Puzzle-Solving. In Exploration-based Models of Design and Search-Based Models of Design, AI in Design ’92. Pittsburg, PA: Carnegie-Mellon University.

Thórisson, K. R., 2012. A New Constructivist AI: From Manual Construction to Self-Constructive Systems. In P. Wang and B. Goertzel (eds), Theoretical Foundations of Artificial General Intelligence. Atlantis Thinking Machines, 4:145-171.

Thórisson, K. R., Bieger, J., Thorarensen, T., Sigurðardóttir, J.S., Steunebrink, B. R., 2016a. Why Artificial Intelligence Needs a Task Theory: And What It Might Look Like. In Proc. Artificial General Intelligence (AGI-16), 118–128. https://doi.org/10.1007/978-3-319-41649-6_12 - accessed Feb. 13, 2024.

Thórisson, K. R., D. Kremelberg, B. R. Steunebrink, E. Nivel 2016b. About understanding. In B. Steunebrink et al. (eds.), Proc. 9th International Conference on Artificial General Intelligence (AGI-16), 106-117.

Thórisson, K. R., Talbot, A., 2018. Abduction, Deduction & Causal-Relational Models. In Workshop on Architectures & Evaluation for Generality, Autonomy & Progress in AI., International Joint Conference on Artificial Intelligence, Stockholm, Sweden.

Thórisson, K. R., Bieger, J., Li, X. & Wang, P., 2019. Cumulative Learning. Proc. 12th International Conference on Artificial General Intelligence (AGI-19), Shenzhen, China, 198-209.

Thórisson, K. R., 2020. Seed-Programmed General Self-Supervised Learning. Proc. Machine Learning Research, 131:32-70.

Thórisson K. R. & H. Minsky, 2022. The Future of AI Research: Ten Defeasible 'Axioms of Intelligence'. Proc. Machine Learning Research, 192:5-21.

Thórisson, K. R., H. Rörbeck, J. Thompson & H. Latapie, 2023. Explicit Goal-Driven Autonomous Self-Explanation Generation. Proc. Artificial General Intelligence Conference, 286-295.

Venkataraman, S., Chakrabarti, A., 2010. An Integrated Model of Designing. J. Comput. Inf. Sci. Eng. 10. https://doi.org/10.1115/1.3467011 - accessed Feb. 13, 2024.

Wang, P., 2006. *Rigid Flexibility: The Logic of Intelligence*. Springer Applied Logic Series (APLS), vol. 34.

Wang, P. & X. Li, 2016. Different Conceptions of Learning: Function Approximation vs. Self-Organization. Proc. Artificial General Intelligence (AGI 2016), 140–149.

Weimer, D., B. Scholz-Reiter & M. Shpitalni, 2016. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals, **65**(1):417-420.

Wing, J. 2021. Trustworthy AI. Communications of the ACM, **64**(10):64-71.

Žavbi, R., Duhovnik, J., 2000. Conceptual design of technical systems using functions and physical laws. AI EDAM 14, 69–83. https://doi.org/10.1017/S089006040014106X – *accessed Feb. 13, 2024.*