174 OPC Open Peer Commentary on "Computational Constructivist Model as an Anticipatory Learning Mechanism for Coupled Agent-Environment Systems" by Filipo Studzinski Perotto, 2013, *Constructivist Foundations*, **9**(1):46-56.

Constructivist Foundations, **9**(1): 59-61.

The Power of Constructivist Ideas in Artificial Intelligence

Kristinn R. Thórisson • Reykjavik University, Iceland

Upshot • Mainstream AI research largely addresses cognitive features as separate and unconnected. Instead of addressing cognitive growth in this same way, modeling it simply as one more such isolated feature and continuing to uphold a wrong-headed divide-and-conquer tradition, a constructivist approach should help unify of many key phenomena such as anticipation, self-modeling, life-long learning, and recursive self-improvement. Since this is likely to result in complex systems with unanticipated properties, all cognitive architecture researchers should aim to implement their ideas in full as running systems, to be verified by experiment. Perotto's paper falls short on both these points.

§1 Cognitive growth, self-inspection, anticipation (prediction based on partial observation), self-organization – what do these have in common? They are all part of a growing set of concepts from biology, cognitive science, artificial intelligence, and psychology that must be related to one another if we are ever to produce a coherent theory of intelligence, whether in machines, animals, or humans. And if our aim is to build working systems – if our stance is a software engineering one with an end-goal of building deployable systems that can operate in real-world environments, whether it be space probes, house cleaning robots, deep-sea explorers, or stock-market investment programs – then our methodological approach must embody principles that are useful for steering our efforts when designing, architecting, implementing, and testing our systems.

§2 Filipo Perotto presents in his paper a model of an anticipatory learning mechanism, CALM, which is based on constructivist principles. His high-level model of agent-environment coupling, CAES, seems a reasonable one. Both models are based on the fundamental assumptions, which I agree with, that: (a) to understand intelligent behavior we must include in our analysis the context in which it operates; and (b) most environments of any interest to intelligent beings contains a mixture of deterministic and non-deterministic causal connections, with many of the former remaining invisible. In my view, and it would seem Perotto's as well, an environment with complex causal relationships (e.g., our everyday world) gives rise to a vast number of potentially observable phenomena, many of which do not clearly or readily convey their underlying causes; this set of potential observable and inspectable phenomena is nevertheless the only information that an intelligent system has access to, via their sensory apparati, for anticipating how their external environment behaves, so as to efficiently and effectively achieve its goals within it.

§3 Before continuing with direct commentary, some points are in order to elucidate the context in which I look at systems engineering, architecture, and constructivism. Due to the high number of combinatorics that a complex environment will produce, through countless interactions between its numerous elements, an agent must create *models* that isolate and capture some essence of underlying causes (invariants or partial invariants) in this environment (Conant & Ashby 1970). Such models must be capable of representing abstract levels of detail that can be used to steer the operations of a system towards efficient expenditure of computational resources – as any thought spent on details completely unrelated to goals (future and present) would be a waste of the agent's time. Thus, the partial models of the environment that an intelligent agent creates will likely form some sort of a cognitive "random-access" abstraction hierarchy; depending on the type of current goal and situation, the agent can then choose models at a particular level of abstraction at any time to help it exclude irrelevant issues from consideration when decisions are being made about how to achieve the goal in that situation. A coherent, unifying model of cognition following constructivist principles must explain how this works, in particular how goals, models, experiences, and iterative knowledge acquisition and improvement operates in concert to achieve cognitive growth in an agent. An engineering methodology for how to build artificial

systems implementing such functions must go further, by helping with defining specifications for an implementable architecture, and providing guidelines on how to implement them in a way that allows experimental evaluation.

§4 An artificial system built to achieve general intelligence must be able to deal with novel situations – situations not foreseen by its programmers. Instead of being given pre-programmed algorithms by its designers, known to be applicable to particular and specific problems, tasks, situations, or environments, the AI itself must be imbued with the ability to generate algorithms (or: compute a control function – I do not distinguish between the two here). For this to be possible the system must furthermore be equipped with the ability to (re-)program itself, otherwise it cannot sensibly change its own operation in any meaningful way based on acquired experience. And to be able to do so, the system must be reflective – that is, the system's architecture and operational semantics must be represented in a way that enables it to read and interpret its own structure and operation. This is what I consider the essence of a constructivist AI methodology: Specifications for how to imbue machines with the capability to make informed changes (whether slowly or quickly) to their own operation, via the runtime principles embodied in their architecture. I do not believe constructivist AI can be done without some form of self-programming on part of the machine, which in turn cannot be achieved without transparency of its operational semantics. In fact, even more radically, I suspect artificial general intelligence cannot be achieved *at all* without such capabilities; higher levels of cognitive operation in the context of novel or unanticipated tasks, situations, and environments, must require some sort of cognitive growth – namely, some form of re-programming of the cognitive system's operation. Conversely, constructivist views on cognition are so different and incompatible with standard software engineering methodologies, especially with its tradition of manual software creation, that they cannot be used at all for engineering such systems. To address constructivist principles head-on in a computational framework will require a new *constructivist AI methodology* (CAIM; Thórisson 2012).

§5 Whether or not Perotto agrees with my views on the nature and need for constructivist development principles thus outlined, he does make some claims to taking steps toward computational implementations of constructivist principles. In this context many important questions come to mind – chief among them being how effective the ideas are for explaining cognitive growth in nature, and how useful might they be for helping implement artificial general intelligence. As Perotto's paper seems to be aimed more at the second topic, we can ask, firstly, do the ideas presented in his paper help with – or are they likely to lead us to – better software engineering methods for implementing constructivist learning in deployed systems; secondly we can ask, if they do in fact offer some new insights to this end, how much still remains to be explained for such systems to spring forth as a result – or conversely, how big a part of the constructivist puzzle does the work attempt to address? Let's look at these in order.

§6 The aim of AI is not just to speculate but to build working, implemented systems. In AI, any theoretical construct aimed at advancing our understanding of how to implement cognitive functions should ultimately be judged on whether actual implementation can conclusively, or partially, allow us to conclude through reliable means (i.e. scientific experimentation), that the ideas, when operating in a relatively complete AI architecture situated in a complex world (Perotto's target environments), are capable of *scaling up*. By "scaling" I mean the ability for a system to grow in a way that supports recursive self-improvement in in complex environments (e.g., the physical world), with respect to its top-level goals. This question is of course difficult to answer, whether experimentally or analytically. A quick walk down memory lane reminds us, however, that the history of AI is replete with examples of proposals that looked great on paper but completely failed such scale up when implemented in a running system, or when attempts were made to expand the models the ideas embodied to include more of the many functional characteristics that they originally left untouched. Unfortunately, experimental evaluation of Perotto's proposed ideas is given short shrift in the paper, and the support provided to answer this question is inconclusive at best. On this count, therefore, not much can be said about the scalability of Perotto's ideas. This is disappointing because a fundamental feature of known constructivist systems in nature is their capability to grow cognitively with experience – itself a form of scaling-up. Other phenomena, such as the power of the CALM schema formalism to produce new knowledge of complex environments, to support models of self (required for any system capable of self-directed cognitive growth), and their ability to support self-inspection, are also not addressed to any sufficient extent in the work. Since these issues are briefly touched on or left unmentioned, we can only assume that they remain unaccounted for by the present work.

§7 My second question regards the "size of the intelligence puzzle" addressed: A cognitive system must, to have a chance at becoming a comprehensive theory of the major facets of intelligence, include a large number of functions that allows systems built to operate relatively autonomously in complex

environments (e.g., the physical world); this *theoretical scalability* of an isolated mechanism is its perseverance and robustness in light of inclusion in a better (larger, more comprehensive) model/theory, which can in turn serve as the foundation for building systems with increased operating power, including an increased capacity for cognitive growth and architectural complexity. If Perotto's work turns out to be correct, if it indeed offers "steps towards computational implementation of constructivist principles", how much of the phenomenon in question – cognitive growth – remains to be explained? The lack of a clear connection between his CALM and CAES models is already a sign that some amount of work remains to be done in this direction. My own list of candidate principles and features (cf. some already mentioned above) that should be accounted for in any reasonable theory of cognitive growth is, unfortunately, quite a bit longer than that addressed in Perotto's paper. Firstly, as described above, cognitive growth requires some kind of autonomic, recursive self-improvement. Although my team has made some progress on this front recently (Nivel & Thórisson 2013, Nivel et al. 2013), research on the topic is still in its infancy, with a host of unanswered practical and theoretical questions, including: What kind of representations are amenable to automatic self-programming for cognitive growth (existing programming languages and paradigms created for humans require human-level intelligence to be used – which calls for the very phenomenon we are striving to understand how to implement); how to achieve the transparent operational semantics needed for automatic programming, and related to that: how to measure a system's operational semantics; what kind of meta-level control structures can be used to steer cognitive growth; what kinds of control architectures can serve as host architectures for the proposed (or any other) constructivist principles, to name a few of many. Questions regarding theoretical scalability issues loom large.

§8 These are, of course, not simple topics. Quite the contrary, they are deep and challenging. But they are central to constructivist approaches, developmental robotics, and principles of cognitive growth, and it is precisely for that reason that they must not be left unaddressed, lest our efforts become victims to the same oversimplification and incorrect application of divide-and-conquer methodology that has plagued much of AI research in the past half century (cf. Thórisson 2013). Unlike so many other phenomena in AI, e.g., planning, vision, reasoning, and learning, that have been largely addressed by calling them "computational" and studying them in isolation through the same strictly allonomic methodologies as used for banking systems, word processors, and Web page construction, a constructivist methodology holds a promise – a potential – to unify a host of complex cognitive mechanisms, most of which have eluded scientific explanation so far. A holistic stance is by far the most likely to lead to an understanding of the phenomenon of intelligence, and anyone with a constructivist mindset has already taken an important step in that direction. But for this to pave the way towards a better theory a genuine attempt must be made to weave as many key cognitive phenomena into the account as possible; attempt to provide a unifying account. And for any engineering effort to be taken seriously, the requirement for experimental evaluations of (physical and/or virtual) running software systems cannot go ignored. Perrotto's stance on these pressing issues remains for the time being largely unknown; we can only hope that he addresses them in the future.

References

Conant, R. C. & W. R. Ashby (1970). Every Good Regulator of a System Must be a Model of That system. *Int. J. Systems Sci.*, **1**(2):89-97.

Nivel E. & Thórisson K. R. (2013) Towards a programming paradigm for control systems with high levels of existential autonomy. In: Kühnberger K.-U., Rudolph S. & Wang P. (eds.) Artificial general intelligence, 78–87. Springer, Berlin.

Nivel E., Thórisson K. R., Dindo H., Pezzulo G., Rodriguez M., Corbato C., Steunebrink B., Ognibene D., Chella A., Schmidhuber J., Sanz R. & Helgason H. P. (2013) Autocatalytic Endogenous Reflective Architecture. Reykjavik University School of Computer Science Technical Report, RUTR-SCS13002.

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. (2013). Reductio ad Absurdum: On Oversimplification in Computer Science and its Pernicious Effect on Artificial Intelligence Research. In K-U Kühnberger, S. Rudolph & P. Wang (eds.), Proceedings of Artificial General Intelligence (AGI-13), Formal MAGIC – Workshop on formalization in artificial intelligence, Beijing, China, July 31st.

The author. Kristinn R. Thórisson has been doing research in artificial general intelligence and real-time interaction for over two decades, in academia and industry. His AERA constructivist cognitive architecture is the world's first system that can learn complex skills by observation in largely underspecified circumstances. He is a two-time recipient of the Kurzweil Award. Kris has a Ph.D. from Massachusetts Institute of Technology.

Received: 16 October 2013 • Accepted: 24 October 2013