awareness magazine
SELF-AWARENESS IN AUTONOMIC SYSTEMS
10.2417/3201111.003951

Self-adjusting autonomous systems

Michael T. Cox and Don Perlis

A solution to the problem of brittle software systems is to endow them with a metacognitive layer that enables the system to reason about failure.

The long-standing promise of artificial intelligence (AI) is a bright new world in which smart machines positively transform the ways we live and work. However, a well-known difficulty for AI is the problem of brittleness.[1] Autonomous systems tend to 'break' when confronted with even slight deviations from the situations anticipated by their designers. Of course, this is not surprising. Why would something 'work' in a given situation if it was not built to do the right thing? We would not expect Deep Blue (a supercomputer) to be able to play even mediocre checkers, because it was built specifically for chess. But AI strives for systems that work in a variety of situations, including ones not anticipated. The solution that we outline here is to add an additional, metacognitive layer of intelligence that watches a system and adjusts its behaviour when the system fails.

Most approaches attempt to avoid failure altogether by programming specific reactions for various classes of situations, but this requires classes that cover all possible failure states. Unfortunately, the approach has proven intractable. Instead, we note that an intelligent agent learns from a failure. A fool is doomed to repeat it.[2] Metacognition (cognition about cognition) has also been tried before, but much of this research focuses on improving existing performance by adding a metalayer (details published elsewhere[3,4]). Our research is different in that we define a relatively small set of failures or anomalies and provide a generalizable mapping at the metalevel to a small set of recovery response strategies.

A lightweight, three-phase architecture exists for deploying response strategies (see Figure 1). We call this architecture the metacognitive loop or MCL.[5,6] MCL acts as an executive monitor and controller when connected to an intelligent host. First, it *notes* when host behaviour or sensor readings diverge from expectations. Second, it *assesses* any such anomaly and the options the host has for dealing with the difficulty. Finally, it *guides* the host towards putting these options into action.

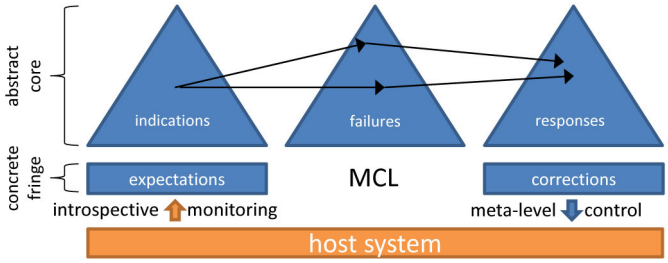

Figure 1. *The metacognitive loop (MCL).*

To implement this architecture, we have developed three ontologies (knowledge representation hierarchies) that support the classification and reasoning abilities in each of the MCL phases. The core of these ontologies contains abstract and domain-neutral concepts. When an actual anomaly is detected, MCL attempts to map it onto the core of the loop so that it may reason about it abstractly. Nodes in the ontologies are linked, expressing relationships between the concepts they represent. There are linkages both within and between the ontologies, which together allow MCL to perform abstraction and reasoning about the anomaly being considered.

Each of the three phases of MCL employs one of the ontologies. The note phase uses an ontology of *indications*, where an indication is a sensory or contextual signifier that the system's expectations have been violated. Processing in the indications ontology allows the assess phase to map nodes in the indications ontology to nodes in the *failure* ontology, which contains nodes that abstractly describe how a system might fail. Nodes in the failure ontology represent the underlying cause of expectation violations. Finally, when hypotheses about the failure have been generated, the guide phase maps that information to its own *response* ontology, which describes means for handling failures at various levels of abstraction.

Reasoning from indications to responses is done by treating the ontologies as a Bayesian network in which all random variables are Boolean.[7] The random variables in the indications ontology are true if the corresponding indication has been observed and are false otherwise. Variables in the failure network

Continued on next page

are true if the corresponding failure has actually occurred and are false otherwise. This is not directly observable, but standard inference methods make it possible to compute a probability distribution over these variables based on the observable evidence (the indications). Finally, random variables in the response ontology are true if the response will likely repair the underlying failure and are false otherwise. Each response has an associated cost, and again standard inference methods are used to find the response with the highest expected utility. Using the lightweight methodology described here and a more knowledge-intensive version described elsewhere,[8] MCL has been shown effective in numerous domains and various cognitive tasks.[5,9,10]

As discussed, our approach to the brittleness problem is to add a metacognitive layer that facilitates a host system's ability to adjust itself in the face of failure. A significant feature that we have started to explore is the use of MCL to decide what, when and how to learn a new skill. Thus, a particular anomaly might lead to the response (conclusion) that training is needed to be better equipped for the given situation. The guide phase then amounts to initiating a training program and monitoring its progress. Finally, we are also exploring a new metacognitive integrated dual-cycle architecture that applies the MCL principles at both the metacognitive and cognitive levels.[11] The hope is that such research will enable a more robust kind of intelligence that can withstand a volatile environment.

This material is based on work supported by the National Science Foundation (grant IIS0803739), the Air Force Office of Scientific Research (grant FA95500910144) and the Office of Naval Research (grant N000140910328).

Author Information

Michael T. Cox
Institute for Advanced Computer Studies
University of Maryland
College Park, MD

Michael T. Cox recently joined the University of Maryland from the Defense Advanced Research Project Agency's Information Innovation Office, where he was a programme manager. Previously, he was a senior scientist at BBN Technologies and an assistant professor of computer science at Wright State University, Dayton (OH). His expertise is in case-based reasoning, computational introspection and mixed-initiative planning.

Don Perlis
Department of Computer Science
University of Maryland
College Park, MD

Don Perlis is professor of computer science. His work is mostly in artificial intelligence: commonsense reasoning; flexible, domain-general, self-adjusting autonomous systems; and philosophical issues surrounding language, mind and consciousness.

References

1. D. Lenat and R. V. Guha, **Building Large Knowledge-Based Systems**, Addison-Wesley, Menlo Park, CA, 1989.
2. M. T. Cox and A. Ram, *An explicit representation of forgetting*, **Proc. Sixth Int'l Conf. Syst. Res. Inf. Cybernet. 2: Adv. AI—Theory Appl.**, pp. 115–120, 1992.
3. Z. B. Rubinstein, S. F. Smith, and T. L. Zimmerman, *The role of metareasoning in achieving effective multiagent coordination*, in M. Cox and A. Raja (eds.), **Metareasoning: Thinking about Thinking**, pp. 217–232, MIT Press, Cambridge, MA, 2011.
4. S. Zilberstein, *Metareasoning and bounded rationality*, in M. Cox and A. Raja (eds.), **Metareasoning: Thinking about Thinking**, pp. 27–40, MIT Press, Cambridge, MA, 2011.
5. M. L. Anderson, S. Fults, D. P. Josyula, T. Oates, D. Perlis, M. D. Schmill, S. Wilson, and D. Wright, *A self-help guide for autonomous systems*, **AI Magazine 29** (2), pp. 67–76, 2008. http://www.agcognition.org/papers/AIMag2008.pdf
6. M. L. Anderson and D. Perlis, *Logic, self-awareness and self-improvement: the metacognitive loop and the problem of brittleness*, **J. Logic Comput. 15**, pp. 22–40, 2005. http://www.agcognition.org/papers/04-33-anderson-perlis.pdf
7. M. Schmill, M. Anderson, S. Fults, D. Josyula, T. Oates, D. Perlis, H. Haidarian, S. Wilson, and D. Wright, *The metacognitive loop and reasoning about anomalies*, in M. Cox and A. Raja (eds.), **Metareasoning: Thinking about Thinking**, pp. 183–198, MIT Press, Cambridge, MA, 2011.
8. M. T. Cox, *Perpetual self-aware cognitive agents*, **AI Magazine 28** (1), pp. 32–45, 2007. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2027/1920
9. M. L. Anderson, T. Oates, W. Chong, and D. Perlis, *The metacognitive loop I: enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance*, **J. Exp. Theoret. Artif. Intell. 18** (3), pp. 387–411, 2006. http://www.agcognition.org/papers/JETAI_final.pdf
10. D. Josyula, **A Unified Theory of Acting and Agency for a Universal Interfacing Agent**, PhD thesis, Department of Computer Science, University of Maryland, College Park, 2005. http://www.cs.umd.edu/~darsana/papers/dissertation/title.html
11. M. T. Cox, T. Oates, and D. Perlis, *Toward an integrated metacognitive architecture*, **AAAI 2008 Symp.**, in press.