The Integration of Cognitive and Metacognitive Processes with Data-driven and Knowledge-rich Structures

Michael T. Cox, Michael Maynord, Tim Oates*,
Matt Paisner, and Don Perlis
University of Maryland, College Park, MD 20742
*University of Maryland, Baltimore County, MD 21250
mcox@cs.umd.edu, maynord@umd.edu, oates@cs.umbc.edu,
mpaisner@umd.edu, perlis@cs.umd.edu

Abstract

This paper examines computational relationships between mind and body and distinguishes thinking about the world from thinking about thinking. The discussion is grounded within the framework of a preliminary computational architecture that proposes an integration of action and perception with cognition and metacognition. We describe the architectural components and discuss the relationship between the meta-level, object level, and ground level. To make this concrete we provide an extended example along with some implemented details.

1 Introduction

Ever since McCarthy originally described the concept of a computer advice taker (McCarthy, 1958), many research projects have embraced the goal of implementing persistent agents that co-exist with people over extended time spans. These agents take various forms including autobiographical agents that have a memory of their own experiences (e.g., Dautenhahn, 1998; Derbinsky & Laird, 2010), social agents that interact and cooperate with humans and other agents (e.g., Breazeal & Scassellati, 1999), and developmental cognitive robots that learn over time (e.g., Weng et al., 2001). Researchers have approached the goal in various ways resulting in theories of human-level intelligence (Cassimatis & Winston, 2004) and artificial general intelligence (Wang & Goertzel, 2012). But currently no research effort has produced a fully robust solution for open worlds and dynamic environments.

Cox (2007) suggests the reason progress on these fronts is difficult is that, despite advances in cognitive systems, few have attempted a full integration of action and perception with both cognition and metacognition. We call such reasoning systems *perpetual self-aware cognitive agents*. Novel implementations exist and have made progress including CogAff (Sloman, 2011), Companion Cognitive Systems (Forbus, Klenk, & Hinrichs, 2009), DIARC (Krause, Schermerhorn, & Scheutz, 2012), EM-One (Singh, 2005), EPILOG (Morbini & Schubert, 2011), INTRO (Cox, 2007), and MCL (Anderson, Oates, Chong, & Perlis, 2006; Schmill et al, 2011). But for practical reasons, most projects emphasize the interaction of at most two of these levels (e.g., the relationship between action and cognition) rather than all three.

This paper examines a cognitive architecture called MIDCA that proposes a computational integration of these three levels. In the next section, we describe MIDCA's basic components followed by a discussion of the relationship between the meta-level and object level. To make this concrete we include an example and then provide further implemented details. We conclude with a brief summary.

2 The MIDCA Architecture

Computational metacognition distinguishes reasoning about reasoning from reasoning about the world (Cox, 2005). As such this assumes a functional approach to philosophy of mind (e.g., Fodor, 1975; Putnam, 1965; Scheutz, 2003). As shown in Figure 1, *the Metacognitive, Integrated, Dual-Cycle Architecture (MIDCA)* (Cox, Oates, & Perlis, 2011) consists of "action-perception" cycles at both the cognitive (i.e., object) level and the metacognitive (i.e., meta-) level. The output side of each cycle consists of intention, planning, and action execution, whereas the input side consists of perception, interpretation, and goal evaluation. A cycle selects a goal and commits to achieving it. The agent then creates a plan to achieve the goal and subsequently executes the planned actions to make the domain match the goal state. The agent perceives changes to the environment resulting from the actions, interprets the percepts with respect to the plan, and evaluates the interpretation with respect to the goal. At the object level, the cycle achieves goals that change the environment (i.e., ground level). At the meta-level, the cycle achieves goals that change the object level. That is, the metacognitive "perception" components introspectively monitor the processes and mental state changes at the cognitive level. The "action" component consists of a meta-level controller that mediates reasoning over an abstract representation of the object level cognition.

Furthermore, and unlike most cognitive theories, our treatment of goals is dynamic. That is, goals are malleable and are subject to transformation and abandonment (Cox & Veloso, 1998; Talamadupula, et al., 2010). Figure 1 shows *goal change* at both the object level and meta-level as the reflexive loops from goals to themselves. Goals also arise from *sub-goaling* on unsatisfied preconditions during planning (the thin black back-pointing arrows on the left of both cycles). Finally new goals arise as MIDCA detects discrepancies between observations and its expectations. It explains what causes the discrepancy, and generates a new goal to remove the cause (Cox, 2007). This type of operation is called *goal insertion* and is a function of interpretation (see the thin, black arrows on the right).

Goal insertion is a fundamental process in MIDCA and occurs at both the object level and meta-level. At the object level, perception provides observations, and plans from memory provide the expectations. The interpretation process detects discrepancies when observations conflict with expectations. The interpretation process will then explain what caused the discrepancy and will generate a new goal. At the meta-level, monitoring provides the observation (a trace of processing at the object level), and a self-model provides the expectations. Like the object level interpretation process, metacognitive interpretation produces an explanation of why the object-level reasoning fails, and it uses the explanation to generate a learning goal to change the knowledge or reasoning parameters of the object level (Cox & Ram, 1999).

Memory plays a central function in both cognitive and metacognitive processes. Thus, our model includes memory, and all cognitive and metacognitive processes have access to it. Note that although memory is shown as separated into two parts, this is an artifact of the split diagram. For example both the object level and the meta-level can access episodic memory. Memory has both declarative knowledge structures represented with indexed frame-based schemas (see Lee & Cox, 2002) and implicit knowledge contained in distributed representations such as GNG nets (see Shamwell, et al., 2012).

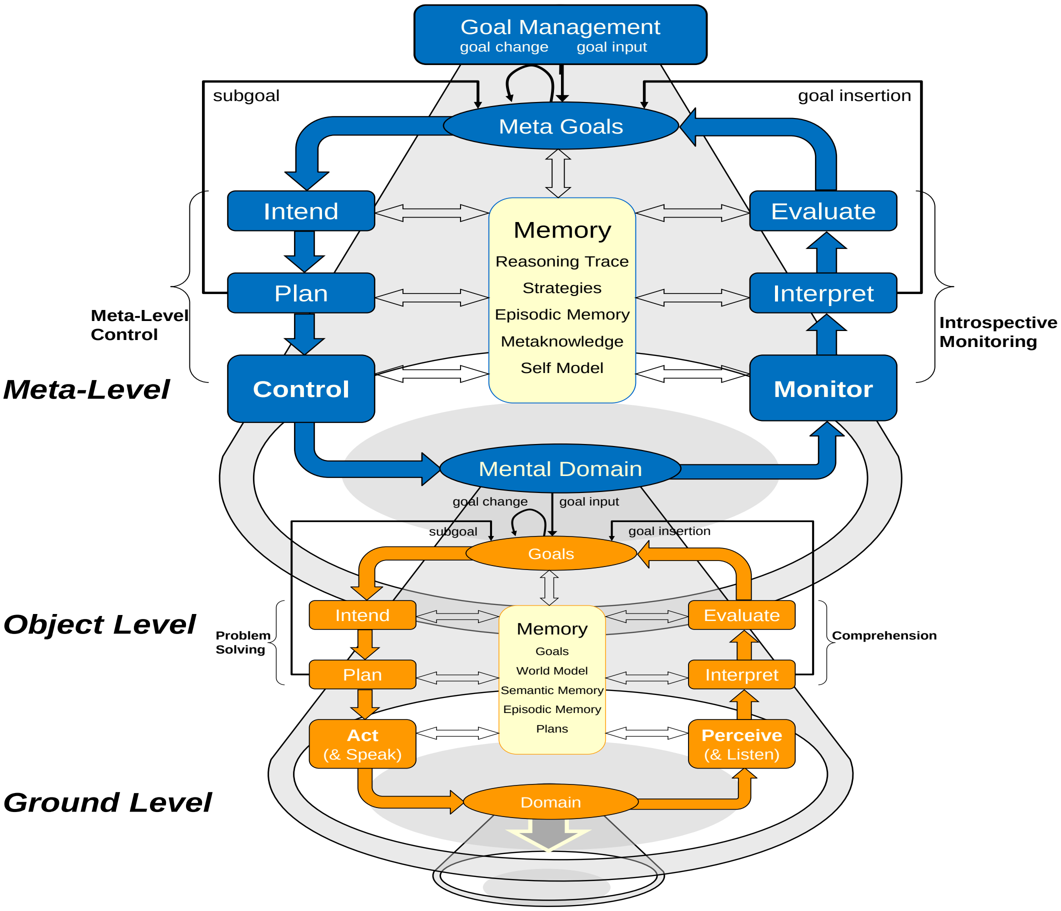


Figure 1. Metacognitive, Integrated, Dual-Cycle Architecture

3 Interaction between Meta and Object Level reasoning

The meta-level can affect the object level in two ways. First the meta-level can act as an executive similar to that of the CLARION cognitive architecture (Sun, Zhang, & Mathews, 2006): deciding between object level parameters; allocating resources between competing object level processes; and setting priorities on object level goals. A qualitatively different approach is for the meta-level to change the structure and content of (object level) reasoning. That is, the meta-level reasoner can change the content of goals, processes, input, or knowledge to orchestrate the object level.[1]

To appreciate the distinctions in the relationship between levels, consider the finer details of the object level as shown in Figure 2. Here the meta-level executive function manages the goal set $\mathcal{G}$. In this capacity, the meta-level can add initial goals (g_0), subgoals (g_s) or new goals (g_n) to the set, can change goal priorities, or can change a particular goal (Δg). In problem solving, the Intend component commits to a current goal (g_c) from those available by creating an intention to perform

[1] An existing meta-level implementation we are using in the early stages of development, Meta-AQUA, has performed this class of operations. For details concerning representations and algorithms implementing this sequence, see Cox & Ram (1999). See Cox (2011) for details of the reasoning-trace representation (i.e., the information passed during metacognitive monitoring).

some *Task* that can achieve the goal (Cohen & Levesque, 1990). The Plan component then generates a sequence of *Actions* (π_k, e.g., a hierarchical-task-net plan, see Nau, et al., 2001) that instantiates that *Task* given the current model of the world (W*) and its background knowledge (e.g., semantic memory and ontologies). The plan is executed by the Act component to change the actual world (W) through the effects of the planned *Actions* (a_i). Problem solving stores the goal and plan in memory to provide the agent expectations about how the world will change in the future. Then given these expectations, the comprehension task is to understand the execution of the plan and its interaction with world with respect to the goal so that success occurs.

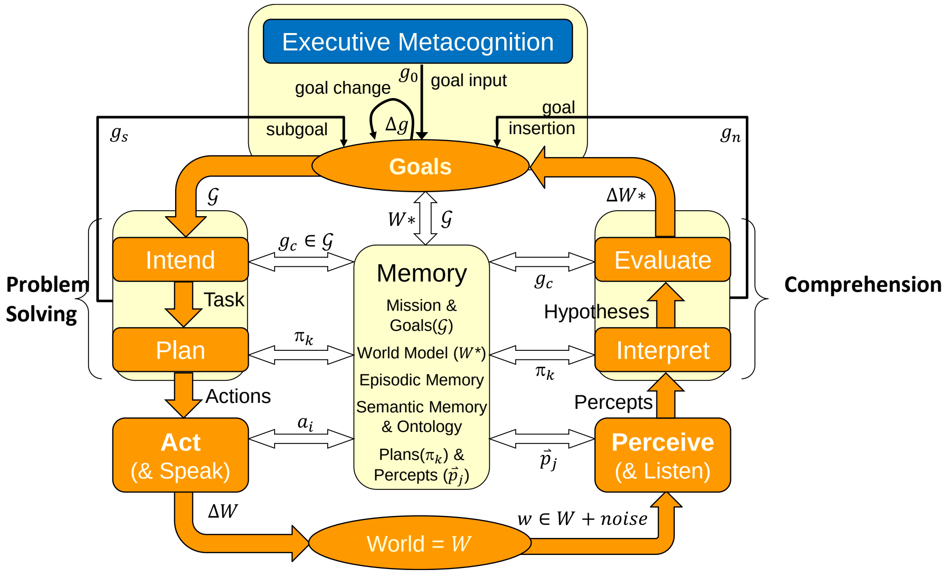


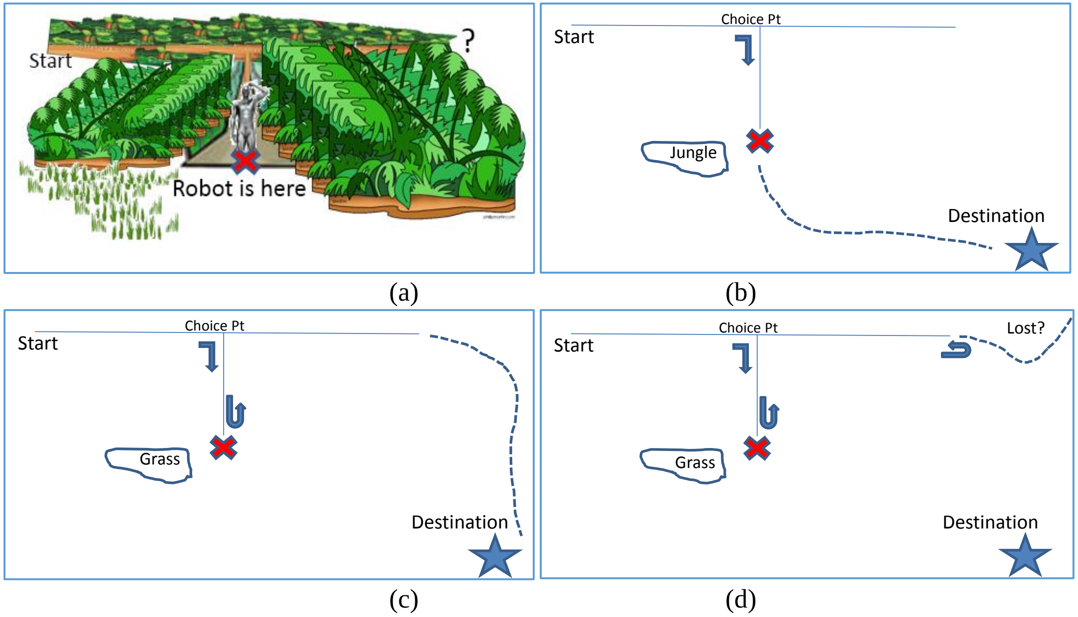
Figure 2. Object level detail with meta-level goal management shown

Comprehension starts with perception of the world in the attentional field via the Perceive component. The Interpret component takes as input the resulting *Percepts* (i.e., $\vec{p}_j$) and the expectations in memory (π_k and g_c) to determine whether the agent is making sufficient progress. A Note-Assess-Guide (NAG) procedure (Anderson & Perlis, 2005; Perlis, 2011) implements the comprehension process. The procedure is to *note* whether an anomaly has occurred; *assess* potential causes of the anomaly by generating *Hypotheses*; and *guide* the system through a response. Responses can take various forms, such as (1) test a Hypothesis; (2) ignore and try again; (3) ask for help; or (4) insert another goal (g_n). Otherwise given no anomaly, the Evaluate component incorporates the concepts inferred from the *Percepts* thereby changing the world model ($\Delta W*$), and the cycle continues. This cycle of problem-solving and action followed by perception and comprehension functions over discrete state and event representations of the environment.

The meta-level performs similar computations. However instead of manipulating declarative representations of ground level states and events, it reasons over traces of object level mental states and mental processes. The trace is provided to an introspective monitoring process, and a meta-level control process manages the goal set $\mathcal{G}$. To make this more concrete, we will examine a simple notional example.

4 Motivational Example: Lost in the jungle

Consider a delivery task carried out in unfamiliar jungle terrain (see Figure 3). Among its daily goals, $\mathcal{G}$, an agent has the objective to take supplies to a training base from a forward depot using an unreliable terrain map. The Intend process chooses from $\mathcal{G}$ the deliver-supply goal to serve as the current goal, g_c, and it selects a *Task* that achieves the goal and passes it to the Plan component. The planner refines the *Task* and generates a sequence of *Actions* that constitute the plan π_k consisting of loading the supplies, departing the depot, finding the destination, and unloading the supplies when there. Now the specific actions in the plan, a_i, are incrementally executed until the agent reaches the goal or detects a problem. Actions such as leaving the depot have associated expectations such as being out of sight of the depot within fifteen minutes. When the comprehension process detects the condition given percepts, $\vec{p}_j$, the action is considered completed.



(a) (b)

(c) (d)

Figure 3. Behavioral, cognitive and metacognitive examples: (a) example illustration; (b) ground-level success; (c) object level expectation failure; and (d) meta-level expectation failure.

In a successful plan, the actions achieve conditions in the world necessary for goal achievement. As shown in panel (b) of Figure 3, the map showed jungle conditions that set up expectations that were confirmed as the plan was executed. When the robot reached the location shown in the red "X," the agent expected jungle and encountered jungle. The Interpret process confirmed these expectations, and the agent reached the destination whereby it unloaded the supplies. At this point the Evaluate process determines that the goal was achieved and the intention is released.

Now in panel (c), we see the condition graphically illustrated in panel (a). The robot expects jungle and instead observes an open field of grass. This is inconsistent with information on the map, and thus constitutes a *contradiction* anomaly at the object level of reasoning.[2] Using the anomaly as a cue, the agent retrieves an explanation from memory and applies it to the situation. The agent thereby assesses the situation by concluding that it took a wrong turn at the previous intersection. The mechanics of this involves reasoning about the action alternatives at the ground level. The planning search tree represents various branches of action choices in the path to the destination. The

[2] Anomaly types also include impasse, unexpected success, false expectation, and surprise (Cox & Ram, 1999).

intersection labeled “Choice Pt” is such a branch point in the search tree. The response is to return to the intersection and take the alternative choice of going forward from there to the destination. This response is a plan change rather than a goal change.[3]

The panel (c) version of the example illustrates the NAG procedure at the object level. Panel (d) is a case of a meta-level invocation of the procedure. Here the expectation is that the explanation from (c) is correct. However if that was so, then the agent would have been at the destination by now. The agent thus concludes that the explanation from (c) is incorrect, and instead the explanation is that the expectation from the map (i.e., that the grassy location should be jungle) was incorrect. This conflict poses a new contradiction (between explanations) and thus an anomaly in the object level as opposed to the ground level. Assessment here involves reasoning about explanation failure. The failure might have occurred due to the agent focusing on the wrong aspects of the input and hence a poor focus of attention. The response is to alter the agent’s model of itself, concluding that it is not expert enough for such tasks and possibly refusing such missions in the future until it possess further experience.

Currently MIDCA is in a preliminary stage of development, and only parts of this scenario are now possible. However many components function as described although not as of yet within an integrated whole. The following section identifies some of the implemented parts during a discussion of bottom-up and top-down aspects of the architecture.

5 D-track and K-track Processes in Comprehension

The NAG procedure at both meta- and object levels has two variations that represent a bottom-up, data-driven track and a top-down, knowledge rich, goal-driven track (c.f., CLARION). The data-driven track we call the *D-track*; whereas the knowledge rich track we call the *K-track*. The D-track is partially implemented as a Bayesian network of ontologies (Schmill, et al., 2011) and partially by a GNG (growing neural gas) network of proto-concepts. The K-track as it currently exists is implemented as a case-based explanation process (Cox & Burstein, 2008).

The representations for expectations significantly differ between the two tracks. K-track expectations come from explicit knowledge structures such as action models used for planning and ontological conceptual categories used for interpretation. Predicted effects form the expectations in the former and attribute constraints constitute expectation in the latter. D-track expectations are implicit by contrast. Here the implied expectation is that the probabilistic distribution of observations will remain the same. When statistical change occurs instead, an expectation violation is raised.

The D-track NAG procedure uses a novel approach for noting anomalies. We apply a statistical metric called the *A-distance* to streams of predicate counts in the perceptual input. This enables MIDCA to detect regions whose statistical distributions of predicates differ from previously observed input (Cox, Oates, Paisner, & Perlis, 2012). These regions are those where change occurs and potential problems exist.

When a change is detected, its severity and type can be determined by reference to a neural network in which nodes represent categories of normal and anomalous states. This network is generated dynamically with the growing neural gas algorithm (Fritzke, 1995) as the D-track processes perceptual input. This process leverages the results of analysis with A-distance to generate anomaly archetypes, each of which represents the typical member of a set of similar anomalies the system has encountered. When a new state is tagged as anomalous by A-distance, it is associated with one of these groups, allowing MIDCA to prioritize explanations and responses that have proven effective with past anomalies in the same category.

[3] This example does not address goal insertion. This would occur if the agent for example discovered an opponent while travelling to the destination. The decision then would be between treating the opponent as a problem or threat to be solved (hence goal insertion might generate a goal to counter the opponent) or as an obstacle to avoid.

Response guiding (in terms of goal insertion) is done through a conjunction of two algorithms both of which work over predicate representations of the world. Tilde (Blockeel, & De Raedt, 1997) is an extension of C4.5, the standard decision tree algorithm, and FOIL (Quinlan, 1990) is a rule generation algorithm producing conjunctions of predicates to match a concept reflected in a training set. Given a world state interpretation, the state is first classified using Tilde into one of multiple scenario classes, where each class has an associated goal generation rule generated by FOIL. Given an interpretation and a class, different groundings of the variables of the FOIL rule are permuted through until either one is found which satisfies that rule (in which case a goal can be generated) or until all permutations of groundings have been attempted (in which case no goal can be generated). This approach to goal insertion is naïve in the sense that it constitutes a mapping between world states and goals which is static with respect to any context; there is no reasoning in this D-track goal generation scheme.

The K-track NAG procedure is presently under development, and we plan to implement a process similar to that used by the Meta-AQUA system (Cox & Ram, 1999) and other case-based interpretation systems. In Meta-AQUA frame-based concepts in the semantic ontology provide constraints on expected attributes of observed input and on expected results of planned actions. When the system encounters states or actions that diverge from these expectations, an anomaly occurs. Meta-AQUA then retrieves an explanation-pattern that links the observed anomaly to the reasons and causal relationships associated with anomaly. A response is then generated from salient antecedents of the instantiated explanation pattern (see Cox, 2007 for details).

One obvious approach to the interaction between the D-track and K-track would be simply to call K-track algorithms only on regions detected by D-track anomaly detection. This would be more efficient because the overhead for the K-track method is greater than that of the A-distance method. But more nuanced approaches exist. For instance the weight of one procedure over the other may be a function of features including resources available and factors such as urgency. Many other issues remain to be examined in detail. These include the decision between plan change (as in the jungle example) and goal change and the allocation of responsibility for this decision between meta-level goal management and the Intend component.

6 Conclusion

This paper introduced the MIDCA architecture as a candidate framework for future perpetual self-aware cognitive agents. Dual cognitive and metacognitive action-perception cycles were described and related to an extended notional example. Implementation was distinguished from design goals. This work represents a unique effort to implement a full integration of action, perception, cognition, and metacognition.

Acknowledgments

We thank the reviewers for their insights and suggestions. This material is based upon work supported by ONR Grants # N00014-12-1-0430 and # N00014-12-1-0172 and by ARO Grant # W911NF-12-1-0471.

References

Anderson, M. L., & Perlis, D. (2005). Logic, self-awareness and self-improvement: The metacognitive loop and the problem of brittleness. *Journal of Logic and Computation 15*(1).

Anderson, M. L., Oates, T., Chong, W., & Perlis, D. (2006). The metacognitive loop I: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. *Journal of Experimental and Theoretical Artificial Intelligence 18*(3), 387-411.

Blockeel, H., & De Raedt, L. (1997). *Experiments with top-down induction of logical decision trees*. Technical Report CW 247, Dept. of Computer Science, K.U.Leuven.

Breazeal, C., & Scassellati, B. (1999). How to build robots that make friends and influence people. *1999 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS-99). Kyongju, Korea.

Cassimatis, N., & Winston, P. (Eds.) (2004). Achieving human-level intelligence through integrated systems and research: Papers from the AAAI fall symposium. Technical Report FS-04-01. Palo Alto, CA: AAAI Press.

Cohen, P. R. & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence 42*(2-3), 213 – 261.

Cox, M. T. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence 169* (2), 104-141.

Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI Magazine 28*(1), 32-45.

Cox, M. T. (2011). Metareasoning, monitoring, and self-explanation. In M. T. Cox & A. Raja (Eds.), *Metareasoning: Thinking about thinking* (pp. 131-149). Cambridge, MA: MIT Press.

Cox, M. T., & Burstein, M. H. (2008). Case-based explanations and the integrated learning of demonstrations. *Künstliche Intelligenz (Artificial Intelligence) 22*(2), 35-38.

Cox, M. T., Oates, T., Paisner, M., & Perlis, D. (2012). Noting anomalies in streams of symbolic predicates using A-distance. *Advances in Cognitive Systems 2*, 167-184.

Cox, M. T., Oates, T., & Perlis, D. (2011). Toward an integrated metacognitive architecture. In P. Langley (Ed.), *Advances in Cognitive Systems, papers from the 2011 AAAI Symposium* (pp. 74-81). Technical Report FS-11-01. Menlo Park, CA: AAAI Press.

Cox, M. T., & Ram, A. (1999). Introspective multistrategy learning: On the construction of learning strategies. *Artificial Intelligence 112*, 1-55.

Cox, M. T., & Veloso, M. M. (1998). Goal transformations in continuous planning. In M. desJardins (Ed.), *Proceedings of the 1998 AAAI Fall Symposium on Distributed Continual Planning* (pp. 23-30). Menlo Park, CA: AAAI Press.

Dautenhahn, K. (1998). Meaning and embodiment in life-like agents. In C. Nehaniv (Ed.), *Plenary Working Papers in Computation for Metaphors, Analogy and Agents* (pp. 24-33). University of Aizu Technical Report 98-1-005.

Derbinsky, N., & Laird, J. E. (2010). Extending soar with dissociated symbolic memories. In Mei Y. Lim & W. C. Ho (Eds.), *Proceedings of the Remembering Who We Are Human Memory for Artificial Agents Symposium,* at the AISB 2010 convention, 29 March 1 - April 2010, De Montfort University, Leicester, UK.

Fodor, J. (1975). *The language of thought*. Cambridge, MA: The MIT Press.

Forbus, K., Klenk, M., & Hinrichs, T. (2009). Companion cognitive systems: Design goals and lessons learned so far. *IEEE Intelligent Systems 24*, 36–46.

Fritzke, B. (1995). A growing neural gas network learns topologies. In Tesauro, G., Touretzky, D. S., & Leen, T.K. (Eds.), *Advances in Neural Information Processing Systems 7*. Cambridge, MA: MIT Press.

Krause, E., Schermerhorn, P., & Scheutz, M. (2012). Crossing boundaries: Multi-level introspection in a complex robotic architecture for automatic performance improvements. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.

Lee, P., & Cox, M. T. (2002). Dimensional indexing for targeted case-base retrieval: The SMIRKS system. In S. Haller & G. Simmons (Eds.), *Proceedings of the 15th International FLAIRS Conference* (pp. 62-66). Menlo Park, CA: AAAI Press.

McCarthy, J. (1959). Programs with common sense. In *Symposium Proceedings on Mechanisation of Thought Processes* (Vol. 1, pp. 77-84). London: Her Majesty's Stationary Office.

Morbini, F., & Schubert, L. (2011). Metareasoning as an integral part of commonsense and autocognitive reasoning. In M. T. Cox & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp. 267-282). Cambridge, MA: MIT Press.

Nau, D., Muñoz-Avila, H., Cao, Y., Lotem, A., & Mitchell, S. (2001). Total-order planning with partially ordered subtasks. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (pp. 425-430). Menlo Park, CA: International Joint Conference on Artificial Intelligence, Inc.

Perlis, D. (2011). There's no 'me' in meta - or is there? In M. T. Cox & A. Raja (Eds.), *Metareasoning: Thinking about thinking* (pp. 15-26). Cambridge: MIT Press.

Putnam, H. (1965). Brains and behavior. In J. Butler (Ed.), *Analytical philosophy*. (Second Series) Oxford: Blackwell.

Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning 5*, 239-266.

Scheutz, M. (Ed.) (2003) *Computationalism: New directions*. Cambridge, MA: The MIT Press.

Schmill, M., Anderson, M., Fults, S., Josyula, D., Oates, T., Perlis, D., Shahri, H., Wilson, S., & Wright, D. (2011). The metacognitive loop and reasoning about anomalies. In M. T. Cox and A. Raja eds., *Metareasoning: Thinking about thinking* (pp. 183-198). Cambridge, MA: MIT Press.

Shamwell, J., Oates, T., Bhargava, P., Cox, M. T., Oh, U., Paisner, M., & Perlis, D. (2012). The robot baby and massive metacognition: Early steps via growing neural gas. In *Proceedings of the IEEE Conference on Development and Learning - Epigenetic Robotics 2012* (ICDL/EpiRob). Los Alamitos, CA: IEEE.

Singh, P. (2005). *EM-ONE: An architecture for reflective commonsense thinking*. Ph.D. dissertation. Electrical Engineering and Computer Science. MIT. Boston.

Sloman, A. (2011). Varieties of meta-cognition in natural and artificial systems. In M. T. Cox & A. Raja (Eds.), *Metareasoning: Thinking about thinking* (pp. 307–323). Cambridge, MA: MIT Press.

Sun, R., Zhang, X., & Mathews, R. (2006). Modeling meta-cognition in a cognitive architecture. *Cognitive Systems Research 7*(4), 327-338.

Talamadupula, K., Benton, J., Schermerhorn, P., Kambhampati, S., Scheutz, M. (2010). Integrating a closed world planner with an open world robot: A case study. In *Proceedings of AAAI 2010*. Palo Alto: AAAI Press.

Wang, P., & Goertzel, B. (2012). *Theoretical foundations of artificial general intelligence*. Berlin: Springer.

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., & Thelen, E. (2001). Autonomous mental development by robots and animals. *Science 291*, 599-600.