Toward an Integrated Metacognitive Architecture

Michael T. Cox, Tim Oates* and Don Perlis

University of Maryland, College Park, MD 20742

University of Maryland, Baltimore County, MD 21250*

mcox@cs.umd.edu; oates@cs.umbc.edu; perlis@cs.umd.edu

Abstract

Researchers have studied problems in metacognition both in computers and in humans. In response some have implemented models of cognition and metacognitive activity in various architectures to test and better define specific theories of metacognition. However, current theories and implementations suffer from numerous problems and lack of detail. Here we illustrate the problems with two different computational approaches. The Meta-Cognitive Loop and Meta-AQUA both examine the metacognitive reasoning involved in monitoring and reasoning about failures of expectations, and they both learn from such experiences. But neither system presents a full accounting of the variety of known metacognitive phenomena, and, as far as we know, no extant system does. The problem is that no existing cognitive architecture directly addresses metacognition. Instead, current architectures were initially developed to study more narrow cognitive functions and only later were they modified to include higher level attributes. We claim that the solution is to develop a metacognitive architecture outright, and we begin to outline the structure that such a foundation might have.

Introduction

For many years, the study of metacognition was a relatively isolated and limited field within cognitive science. But recently the subject has gained a more prominent role and attained widespread attention within both psychology and computer science. Studies include a wide ranging set of experiments in the cognitive psychology literature that demonstrate the ability and limitations of human self-monitoring and control of reasoning. For example, Dunlosky and colleagues (see Dunlosky, Serra, & Baker, 2007) have investigated subjects' use of memory judgments to regulate further time spent on learning material for memory tasks. Bogunovich and Salvucci (2011) have shown that subjects can dynamically control their problem-solving process based upon knowledge of the relative task demands and the required mental resources to complete it. Metcalf and colleagues (e.g., Metcalfe, Eich, & Castel, 2010) have studied subjects' ability to discriminate between actions caused by others and by themselves. In the artificial intelligence community, many new research agendas have arisen in recent years under the terms metacognition in computation and metareasoning (see Anderson & Oates, 2007; Cox, 2005 for pertinent reviews). Yet within these disparate research efforts, few take a comprehensive approach or place their claims and findings within the context of a cognitive architecture. Here we intend to address this concern by outlining a new cognitive architecture that focuses on metacognition and that proposes to integrate many related metacognitive phenomena.

Rather that constituting a technical issue at the periphery of intelligence, the problem of reasoning about reasoning is a central challenge at the heart of high-level cognition. Although some disagreement exists about the degree to which metacognition is uniquely human (e.g., Smith et al., 2009), it is nonetheless one of the most prominent aspects of our cognitive make-up. For example, metacognition is particularly associated with critical thinking independent of a more general cognitive aptitude or achievement (Ku & Ho, 2010). But despite the centrality and importance of this phenomenon, few cognitive architectures model it or directly address its computational structure.

A select few research efforts in metacognition do base their models upon a cognitive architecture. For example, the work of Salvucci cited above uses the ACT-R architecture to model the problem-solving behavior in question. However, the modeling has concentrated mainly on threaded multitasking behavior and has not clarified the specific relationship between multitasking and metacognition. Furthermore, the ACT-R theory was not developed with specific metacognitive effects in mind. The control of reasoning is actually considered to be part of the cognitive function; whereas monitoring of reasoning is classified as metacognitive (see for example Anderson, Betts, Ferris, & Fincham, 2011). In any case, many of the existing cogni-

tive architectures have limitations when modeling meta-cognitive activity and have been modified on an as needed basis to fit the circumstances.

To clarify the problem, let us consider the most basic mechanisms of high-level reasoning about the self. Nelson and Narens (1992) first proposed a model of metacognition in humans that divided mental processes into an object level (cognition) and a meta-level (metacognition) such that the meta-level contains within it a dynamic model of the object level. Monitoring is the flow of information from the object level to the meta-level, and control information flows from the meta-level to the object level. It was Cox (2005) that first noted that this model applies to computational models of metareasoning as well as to human models of metacognition. Finally Cox & Raja (2011) extended this model to the one shown in Figure 1.

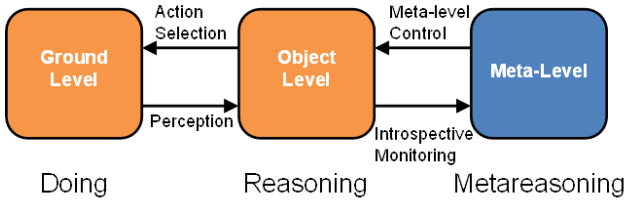

Figure 1. Simple model of metareasoning

The relationship between ground and object levels represents a classical action and perception cycle. Events at the ground level are perceived as a set of percepts through some collection of sensors. In response the object level selects some sequence of actions and performs them through the agent's effectors. This interacts with and changes the environment at the ground level, and the cycle continues. Likewise, the relationship between the object level and the meta-level is cyclical. Like perception, *introspective monitoring* is the reception of information that describes the events and transitions at the object level. In response, *meta-level control* determines a sequence of mental actions or otherwise mediates the functioning of the object level. Metareasoning causes the reasoner to think in certain ways as reasoning causes the agent to act in a certain manner. It is this duality of reasoning and metareasoning (or of cognition and metacognition) [1] that an integrated architecture must capture and through its fixed structure enable the modeling of various tasks both computationally and psychologically.

Note that most existing cognitive architectures concern the orange areas associated with the action and perception cycle. This paper will recast this cycle in a new structural and representational framework based on a model by Norman (1986) and will add the blue colored layer to formulate an analogous cycle at a higher level. We will introduce this framework by first considering in the next section two different systems that model metacognitive processes and that share many aspects. Then in the subsequent section, we will propose a specific new metacognitive-architectural framework. The conclusion will summarize and will suggest an applied, task domain of self-regulated learning within which to implement and evaluate our ideas.

Two Current Implementations

Researchers have implemented a number of computational systems that follow the basic metareasoning model above. Here we will briefly examine two such implementations with which the authors have experience. First we consider the concept and implementation of the Meta-Cognitive Loop and then the Meta-AQUA system.

The Meta-Cognitive Loop

The MetaCognitive Loop (MCL) (Anderson & Perlis, 2005; Schmill et al., 2011) represents an architecture that enables a system to reason about itself. Conceptually, MCL in its current form consists of a module attached to a given "host" system H (see Figure 2). The host system supplies information to MCL about its possible actions and their purposes in the form of general expectations of outcomes of those actions under various conditions. For instance, the action "go to location X" might be associated with the expectation "be at location X within ten seconds." H also supplies MCL with sensor readings, such as "current location" and "time" so that MCL can monitor the success of H's actions by comparing expectations and outcomes. This is the first of three steps in MCL's loop: note anomalies (mismatches between expectations and outcomes).

Once an anomaly is noted, MCL assesses the anomaly in

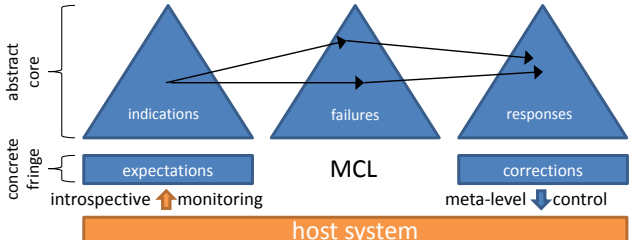

Figure 2. The meta-cognitive loop

light of its knowledge about factors such as importance,

[1] In this paper we use metareasoning and metacognition in a roughly synonymous way. We will often use the term metareasoning when referring to AI aspects and metacognition when referring to human aspects. However we also consider metareasoning to be a more narrow term when considered technically. It means reasoning about reasoning and does not necessarily include related issues such as metaknowledge or reasoning about other agent's reasoning. Metacognition on the other hand is used here and by other authors more generally in a broad sense to include many related high-level functions such as self-modeling, metalinguistic self-referencing, knowing what one does not know, and feelings of self-efficacy.

likely causes, similarity to other anomalies, and possible responses (e.g., try again, ask for help, recalibrate sensors, postpone, give up). Finally, MCL selects – and guides H to enact – a response based in part on its likelihood of success. Thus MCL is in effect the following algorithm: note; assess; guide; repeat. Our underlying vision of MCL, then, is as a module that, when suitably attached to a host system H, leads dynamically to enhanced performance in the MCL+H symbiot.

MCL is intended to be a domain-general framework for autonomously identifying and dealing with anomalous situations. To achieve this goal, MCL must require minimal domain-dependent engineering and therefore must also be able to leverage abstract, domain-independent reasoning to find solutions to problems without burdening the host system designer with the task of specifying how the host might fail and how to cope with those failures. To allow for this ability, we have developed three ontologies (see Figure 2) that support the required classification and reasoning abilities in each of the three MCL phases of note, assess, and guide. The core of these ontologies contains abstract and domain-neutral concepts; when an actual anomaly is detected, MCL attempts to map it onto the MCL core 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.

In our current implementation, each of the three phases of MCL employs one of the ontologies to do its work. 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 indication ontology allows the assess phase to map nodes in the indication 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 type have been generated, the guide phase maps that information to its own response ontology, which describes means for dealing with failures at various levels of abstraction. Through these three phases, reasoning starts at the concrete, domain-specific level of expectations, becomes more abstract as MCL moves to the concept of system failures, and then becomes more concrete again as it selects a specific system response based on the hypothesized failure.

Reasoning from indications to responses is done by treating the ontologies as a Bayesian network in which all random variables are Boolean (Schmill et al., 2011). The random variables in the indications ontology are true if the corresponding indication has been observed and are false otherwise. Random variables in the failure network 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.

MCL in various versions has been used in a wide variety of domains, including robot navigation, reinforcement learning, natural-language human-computer dialog, and playing an arcade tank-game; see Haidarian et al. (2010) for an overview. (Among these, the first two (navigation and RL) were not metacognitive in the strict sense: the implementations did not involve an explicit knowledge base, and the monitoring and control were applied directly to behavior rather than beliefs.)

Each initial study involved a new implementation of the note-assess-guide idea specific to that application. Moreover, there was no clear distinction between the host and the metacognitive module. But once it became clear that similar underlying strategies were effective in different domains, the current general-purpose version – with relatively fixed ontologies of indications, failures, and responses – was designed and implemented.

One contribution of our prior, domain-specific implementations of MCL is the accrual of compelling evidence, given the breadth of the domains and the success of MCL in each case, that the overall approach of expectation-driven metacognition is both general and powerful. More recent contributions include the formulation of metacognition as probabilistic reasoning over ontologies of indications, failures, and repairs, specification of domain-general core ontologies, and development of a concrete implementation of the theory along with well-defined interfaces for connecting arbitrary host systems to MCL.

We are currently testing MCL in a domain that simulates multiple robotic rovers on the surface of Mars that communicate with each other using formal languages and with humans using fragments of English. Each robot's interactions with the domain (Mars) are monitored by MCL, as are the robot's communications with humans. The goal is to test our claims about the generality of MCL, using the same core ontologies and computation in the two very different domains of natural language communication and accomplishing goals of the surface of Mars. Ablation experiments will determine which components of MCL are most important in four experimental conditions: no MCL but with a fixed maintenance schedule; performing all maintenance actions when MCL notes a problem; allowing MCL to choose a single maintenance action when a problem is

noted; full MCL in which the note-assess-guide cycle can repeat as needed.

Meta-AQUA

Meta-AQUA (Cox, 2007; Cox & Ram, 1999) is an implementation of a theory of Introspective Multistrategy Learning (IML) and a cognitive model of introspection. As a model of goal-driven learning it shares much with the note-assess-guide cycle of MCL. For example they both focus on responding to failure. However as a model of introspection, many differences exist between the two.

IML theory focuses on the deliberative aspects of learning as opposed to a more unconscious process of reinforcement or operant conditioning. Like the note-assess-guide cycle, introspective learning starts when expectations fail. A contradiction is said to exist when actual observations or outcomes differ significantly from an agent's expectations. However failure symptoms other than contradiction can also trigger learning. An impasse occurs when no expected outcome can be generated, and a surprise occurs when there is no expectation and no attempt to generate an expectation. All three of these cases are considered anomalies or symptoms of failure in IML theory. They are detected in an identification phase similar to the note step of the metacognitive loop.

Given a failure symptom, the next phase is to explain the failure or the underlying cause. Here explanation can be thought of as a diagnostic symptom to fault mapping and as such is an instance of self-diagnosis. The explanation process is considered a generation phase of learning and corresponds to the assess step in MCL. The task of the phase is to generate an explanation that bridges the gap from failure symptom to failure cause. For example the symptom may have been that a person expects to be correct when calling a new acquaintance Bill instead of John, and the cause could be that he confused John with another person he already knew. Thus the contradiction is between the expectation name=Bill and the observation name=John; whereas, the causal explanation is that the person forgot the correct name (i.e., the cues in the context of the utterance were not sufficient to retrieve the name of John, because the storage indexes did not match the retrieval probe).

The explanation generation phase in IML is a knowledge-rich process. The Bayes net approach in MCL is a knowledge light approach. The explanations in the former contain not only the structure of the failure, but they also identify what possible parts of the structure may be responsible for the failure. For example the name-forgetting explanation may highlight the index of the memory retrieval as the main causal factor in contradictions of this type. Along with this identification, the structure will also associate a learning goal with the flawed index. The goal would be to achieve an index that would have retrieved the correct name given the context. Subsequently a deliberate learning process would create and execute a plan to achieve the learning goal. This plan and execute phase corresponds to MCL's response step.

The performance task of the Meta-AQUA system is story understanding. The system inputs a story in a pre-parsed conceptual representation, and it outputs a model of the actions and events in the story along with a set of explanations that explain why agents perform particular actions. A story is understood if the incremental representation of the story is predictive (i.e., schema instantiations anticipate subsequent events in the story stream), coherent (i.e., the graph structure underlying the representation is fully connected), and explanatory (i.e., annotations provide motivational causality for interesting or anomalous events).

As shown in Figure 3, Meta-AQUA contains a number of major subsystems. At the ground level is a story generation module called Tale-Spin that produces a sequence of stories for Meta-AQUA to understand. The stories are at the ground level because the performance task at the object level is to understand events in the story. One could think of input story concepts as percepts from an imagined environment. The performance subsystem uses knowledge structures from its background knowledge (BK) to interpret and explain events and to create a representation of the story in it foreground knowledge (FK). The FK and BK constitute a memory system for Meta-AQUA.

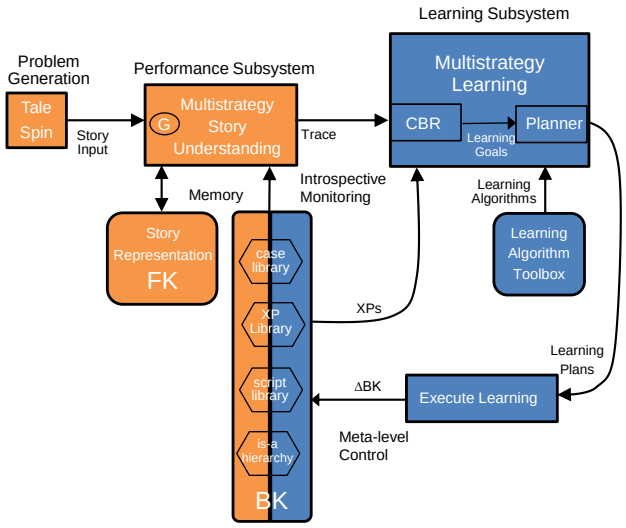

Figure 3. Meta-AQUA cognitive architecture

The Meta-AQUA system implements the IML model as an introspective version of case-based explanation (see for example Schank, Kass & Riesbeck, 1994). That is, the computational problem is to retrieve from a case-base of introspective meta-explanations (Meta-XPs) an explanation pattern that partially matches the structure of the current

trace of reasoning representing a given anomaly. The structures are called meta-explanations, because initially they represented meta-level explanations of object-level explanation failure. The Meta-AQUA program is a model of introspection, because it represents traces of reasoning with Meta-XPs and allows the system to reason about the prior reasoning that underlies this representation.

When explanation failure occurs at the object level task of understanding the story, a trace of the reasoning that preceded the failure is passed to the learning subsystem. This trace constitutes a form of introspective monitoring. The learning component then retrieves an introspective meta-explanation and binds it to the reason trace. As mentioned above, this instantiated explanation pattern will then contain a set of learning goals with which a planner can generate a learning plan. The execution of this plan will then change memory structures in the BK, thus producing meta-level control of the object level reasoner. If learning is successful, similar explanation failures will not reoccur given similar stories in the future. See Cox & Ram (1999) for further details.

One of the key contributions of this research is a representational structure called a *metaexplanation pattern* (Meta-XP). These structures can represent how an explanation is created (i.e., record the reasoning that generates an ordinary explanation), and they can represent causal patterns that explain why explanations fail (i.e., capture past cases of meta-reasoning about failure). In IML theory, the notion of a Meta-XP has been extended to represent performance failure in general, not just explanation failures.

A key construct in the theory and implementation is the idea of a *learning goal*. Rather than specifying a desired state of the world, a learning goal represents a desired state of knowledge. A prototypical example of a learning goal is a question, the answer to which represents the achievement of the goal. A major hypothesis of this work is that learning goals are necessary to mediate between the explanation of failure and the learning needed to avoid the failure; a direct mapping is not sufficient in all cases. Many case-based reasoning systems use the direct indexing of repairs by indexes that represent the conditions under which they are appropriate. This work seeks to demonstrate that such linkage may lead to incorrect results when the chosen learning methods interact. Researchers cannot assume the learning algorithms are independent. Just as planning goals assist in alleviating the problems of interacting planning steps, learning goals can solve the problems of interacting learning strategies.

Limitations and Gaps

Although both MCL and Meta-AQUA have produced significant contributions, neither system implements a full metacognitive architecture. They each contain a number of limitations that prevent them from fully acting as a model of metacognition.

Although MCL when attached to a host such as a robot represents an agent embodied in a real world, MCL has a weak model of perception and no model of high-level understanding and interpretation. Meta-AQUA performs deep understanding and reasons about the mental states of other characters, but it is essentially a disembodied agent, lacking a model of action and personal agency. Whereas Meta-AQUA has a complex multifaceted memory and reasons about memory events, MCL does not have a model of memory or retrieval. However neither agent has an episodic memory to represent cases of personal experience and individual actions. Despite the lack of work on episodic memory in the cognitive architecture literature (but see Nuxoll & Laird, 2007), it has been argued that such autobiographical memory associated with a cognitive and non-verbal self is an important precursor to metacognitive awareness and the development of children's theory of mind (Howe, Courage, & Edison, 2003).

Finally neither MCL nor Meta-AQUA has an explicit model of self. The systems do not have a model of the contents of their background knowledge for example, and thus they cannot answer questions such as what kinds of tasks are they expert at. They have no feelings of confidence as they perform a cognitive task, and thus they cannot decide whether or not they are getting close to an answer. Without such a model the system is severely limited in its ability to explain itself (Cox, 2011). Instead of adding on these missing attributes to an already existing framework, we claim that what is required is a principled new architecture that addresses issues of metacognition from the start.

A Dual-Cycle Metacognitive Architecture

Norman (1986) posits a useful model of human-computer interaction that emphasizes a significant contrast between two major cognitive systems. For him, complex, human interaction is driven by the twin processes of action execution and goal evaluation (see main orange cycle in Figure 4).[2] In the former process, an individual decides to execute actions to achieve goals, and in the latter, the individual evaluates how well the change in the environment matches the goals. Like many cognitive theories, intelligence is organized around an action-perception cycle.

For Norman each part of the cycle contains certain subprocesses. The execution side consists of intention, planning, and then action execution; the evaluation side consists of perception, interpretation, and goal evaluation. The

[2] We are not the first to take inspiration from Norman. Lewis (1998) used the model to more effectively understand various strategies of human-agent interaction.

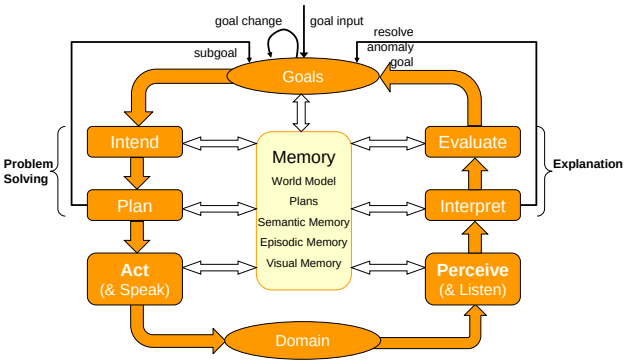

Figure 4. Norman's (1986) model extended

cycle is to take a goal and commit to achieving it in a general way. The human then plans to achieve the goal and subsequently executes the planned actions that change the world, thereby making the goal state so. The evaluation part of the cycle perceives the changes to the world affected by the actions, interprets the percepts with respect to the plan, and evaluates the interpretation with respect to the goal.

The example Norman uses is the goal of having an improved document appearance. Thus a user intends to change the justification from ragged right to justified using a word processor. The plan is to select the text body and then to apply justification formatting. The actions involve a series of mouse movements and key clicks. These operations change the appearance of the document and the user perceives the change. The user interprets the percepts with respect to the plan and then evaluates the document appearance to determine whether the goal is achieved. We embrace this formulation and propose to extend it with a number of embellishments.

One of the important factors that Norman neglected in his model was a role for memory. Many mistakes of reasoning occur due to memory, and it plays a central function in both cognitive and metacognitive processes. We have thus included memory in our model and consider all cognitive processes to have access to it. As seen in Figure 4, memory would contain both semantic and episodic components. Here also would be a model of the current environment, plans, and a visual memory. None of these additions are particularly unique to cognitive architectures, but they are necessary.

Furthermore and like most cognitive theories, Norman's treatment of goals is static. That is, goals are simply given; the model does not account for their genesis. However, as an innovative claim we contend that they are dynamic objects instead and will change over time. Goals are malleable and are subject to transformation and abandonment (Cox & Zhang, 2007). Figure 4 illustrates this in the reflexive loop from goals to themselves. Goals also arise from sub-goaling on unsatisfied preconditions during planning (again see Figure 4). Finally new goals arise as problems are recognized in the environment due to anomalies. The agent recognizes the problem, explains what causes the problem, and generates a new goal to remove the cause (Cox, 2007). This type of operation, called goal insertion, is indicated by the thin, black arrow from the interpretation process in Figure 4.

Taken as a whole, Figure 4 represents the object level and the ground level of a perception-action cycle as shown in Figure 1. Figure 5 shows a metacognitive reflection of Figure 4 integrated into a single, dual-cycle architecture.[3]

Two different ways exist that the meta-level (in blue) can affect the object level (in orange). First the meta-level can act as an executive function. It can decide between object-level parameters, it can allocate resources between competing object-level processes, and it can set priorities on object level goals. The kinds of decisions an executive function may determine are as follows.

- When to switch between cognitive processes.
- When to act at the ground level instead of thinking further.
- When to change goal priorities.
- How resources are distributed between cognitive processes.

Another qualitatively different way in which the meta-level can have an effect on the object level is for it to change the essential structure and content of reasoning. Given that reasoning is goal-directed processing of an input using specific knowledge (Cox & Ram, 1999), the meta-level reasoner can change either the goals, the processes, the input, or the knowledge to orchestrate the object level.

Consider changing the knowledge that an agent uses to make decisions (i.e., learning). As pointed out in an earlier section, Meta-AQUA receives traces of faulty reasoning via introspective monitoring. It then interprets the reasoning trace to explain the failure and generate a learning goal. This goal represents a desired change in its own background knowledge. The system then creates a learning plan and executes that plan to achieve the learning goal. Subsequently this change to its knowledge affects and indirectly controls the subsequent performance at the object level in Meta-AQUA. The above sequence is completely at the meta-level within Figure 5.

The other three strategies are to change the object-level goals of the system, change the processes, or change the input. Given our earlier statement that goals are malleable, we can imagine a meta-level goal management process that

[3] 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. There are not two distinct episodic memory stores.

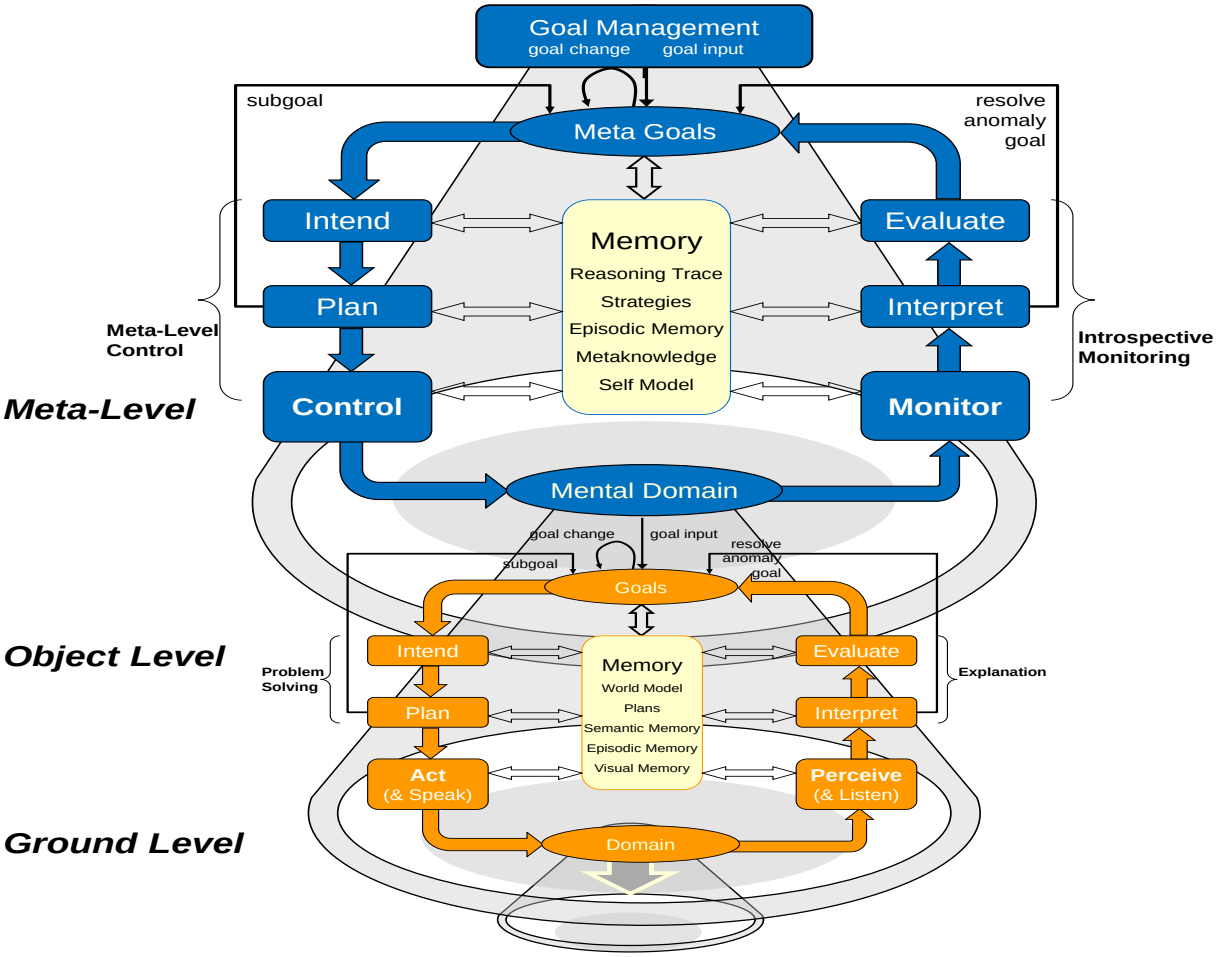

Figure 5. A metacognitive integrated dual-cycle architecture (MIDCA)

changes a checkmate goal in chess into one of achieving a draw. Alternatively the meta-level can select a different process to use for achieving a given goal. For example an agent may switch from using first principles inference to using case-based reasoning to solve a particular problem. Finally the system can change its focus of attention. This could be performed by changing the perceptual field of view (e.g., turning one's head to a different direction).

In an integrated metacognitive architecture, one represents a number of different structures in memory. As shown in Figure 5, an agent has a representation for strategies, an episodic memory for not only what it performed at the ground level but also for what it remembers and thinks (at least to some level of detail), a set of assertions about its own knowledge (or the lack thereof), and a self-model. Given the limits of the size of this paper, we will only state that to be complete, the architecture will provide a mechanism for representing and processing such structures.

Some early work has been performed on parts of this architecture, but most of this model is in the early stages of development. The outline here gives a high-level overview and provides the details for the core of the model.

Conclusion

This paper has proposed a novel architecture that incorporates both a perception-action cognitive cycle and a monitor-control metacognitive cycle. This dual-cycle arrangement includes access to a general memory system and accounts for independent goal-based operations. Having an episodic memory provides an agent with a representation for its past self, having a self-model gives it a representation for its current self, and independent self-generated goals represent its future self.

We plan to further develop the outline presented here by implementing a full architectural specification and applying it to models of self-regulated learning. This task is an important potential application for a number of reasons. First of all much human data has been collected with respect to the task (see for example Azevedo & Cromley, 2004). Secondly the task is wide in scope and very relevant

for the study of metacognition. Students studying for a test must regulate their learning efficiently and at a very high level. They have to not only learn the domain, but they must pace and control themselves using an understanding of what they already know, how well they can perform in the domain, and what is expected of them by the instructor. Therefore the task will combine models of cognition, metacognition, and theory of mind (i.e., models of the mental states of others). Finally self-regulated learning is an important application, because it has the potential of improving the learning ability of actual students. If insights into metacognitive architectures and their implementation can be translated into benefits in the classroom, then the impact will be significant.

Acknowledgments

This material is based upon work supported by NSF Grant # IIS0803739, AFOSR Grant # FA95500910144 and ONR Grant # N000140910328.

References

Anderson, J. R., Betts, S., Ferris, J. L., & Fincham, J. M. (2011).Cognitive and metacognitive activity in mathematical problem solving: Prefrontal and parietal patterns. *Cognitive, Affective, & Behavioral Neuroscience 11*(1):52-67.

Anderson, M. L., & Oates, T. (2007). A review of recent research in reasoning and metareasoning. *AI Magazine, 28*(1): 7-16, 2007.

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).

Azevedo, R., & Cromley, J. G. (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96*(3), 523–535.

Bogunovich, P., & Salvucci, D. D. (2011). The effects of time constraints on user behavior for deferrable interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: CHI 2011*.

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

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

Cox, M. T., & Raja, A. (2011). Metareasoning: An introduction. In M. T. Cox & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp. 3-14). Cambridge, MA: MIT Press.

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

Cox, M. T., & Zhang, C. (2007). Mixed-initiative goal manipulation. *AI Magazine 28*(2), 62-73.

Dunlosky, J., Serra, M. J., & Baker, J. M. C. (2007). Metamemory. In F. Durso (Ed.) *Handbook of Applied Cognition. 2nd Edition*. (pp. 137-161). Chichester, West Sussex, England: John Wiley & Sons, Ltd.

Haidarian, H., Dinalankara, W., Fults, S., Wilson, S., Perlis, D., Schmill, M., Oates, T., Josyula, D., Anderson, M. L. (2010). The metacognitive loop: An architecture for building robust intelligent systems. In *Proceedings of the AAAI Fall Symposium on Commonsense Knowledge*, Arlington, VA, USA, November 11-13, 2010.

Howe, M. L., Courage, M. L., & Edison, S. C. (2003). When autobiographical memory begins. *Developmental Review 23*, 471–494.

Ku, K. Y. L., & Ho, I. T. (2010). Metacognitive strategies that enhance critical thinking. *Metacognition and Learning, 5*(3), 251-267.

Lewis, M. (1998). Designing for human-agent interaction. *AI Magazine, 19*(2), 67-78.

Metcalfe, J., Eich, T. S., & Castel, A. (2010). Metacognition of agency across the lifespan. *Cognition, 116*, 267-282.

Nelson, T. O., & Narens, L., (1992). Metamemory: A theoretical framework and new findings. In T. O. Nelson (Ed.), *Metacognition: Core readings* (pp. 9–24), Boston: Allyn and Bacon. Originally published in 1990.

Norman, D. (1986). Cognitive engineering. In D. Norman & S. Draper (Eds.), *User-centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Nuxoll, A. M., Laird, J. E. (2007). Extending cognitive architecture with episodic memory. In *Proceedings of the twenty-second AAAI conference on artificial intelligence*, AAAI Press, Vancouver, BC.

Schank, R. C., Kass, A., & Riesbeck, C. K. (1994). *Inside case-based explanation*. Hillsdale, NJ: Lawrence Erlbaum Associates.

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 & A. Raja (Eds.) *Metareasoning: Thinking about thinking* (pp. 183-198). Cambridge, MA: MIT Press.

Smith, J. D., Beran, M. J., Couchman, J. J., Coutinho, M. V. C., & Boomer, J. B. (2009). Animal metacognition: Problems and prospects. *Comparative Cognition and Behavior Reviews, 4*, 40–53.