ONR Award #N000141210172 -- Period of Performance: 1 June 2012 through 31 May 2013

MIDCA: A Metacognitive, Integrated Dual-Cycle Architecture for Self-Regulated Autonomy

Computer Science Technical Report No. CS-TR-5025

UMIACS Technical Report No. UMIACS-TR-2013-03

Principal Investigator: Michael T. Cox

mcox@cs.umd.edu

University of Maryland Institute for Advanced Computer Studies

College Park, MD 20742

Subcontractor: Tim Oates

University of Maryland, Baltimore County (UMBC)

Baltimore, MD 21250

Abstract

This report documents research performed under ONR grant N000141210172 for the period 1 June 2012 through 31 May 2013. The goals of this research are to provide a sound theoretical understanding of the role of metacognition in cognitive architectures and to demonstrate the underlying theory through implemented computational models. During the last year, the team has been integrating existing implemented systems to form an initial architectural structure that approximates the major functions of MIDCA. These include the SHOP2 hierarchical planning system and the Meta-AQUA integrated multistrategy learning system. We have also produced substantial progress on the data-driven track of the interpretation procedure. Last year’s work on using the A-distance metric for anomaly detection has been matured, and we have collected substantial observations used in empirical evaluation. Additionally we started implementation of a neural network to induce proto-type nodes for observed anomalies, and we are developing methods to prioritize explanations and responses that have proven effective with past anomalies in proto-type categories. The data are encouraging and the research community has reacted favorably. Several new publications support our claims herein.

Scientific and Technical Objectives

This report documents research performed under ONR grant N000141210172 for the period 1 June 2012 through 31 May 2013. The goals of this research are to provide a sound theoretical understanding of the role of metacognition in cognitive architectures and to demonstrate the underlying theory through implemented computational models.

Across the three year period of performance, this project intends to accomplish the following four objectives. They have not deviated from the original intentions stated in the proposal.

(i) A detailed theoretical understanding of what is required for architectures to be metacognitive;

(ii) A computational framework based upon this theory for implementing such systems;

(iii) A metacognition-enabled robotic platform built upon that framework;

(iv) A detailed empirical evaluation of the implementations and their integration.

Approach

The University of Maryland team is developing a comprehensive theory of cognition and metacognition, constructing a *Metacognitive, Integrated Dual-Cycle Architecture (MIDCA)*, and applying an implementation of this architecture to the domain of robotic mission rehearsal. The dual-cycle architecture integrates a problem-solving and comprehension loop at the object (cognitive) level with a control and monitoring loop at the meta-level. Our approach in this project is to use a standard planner for both planning and control functions and to concentrate upon the comprehension and monitoring processes with parallel knowledge-rich and data-driven approaches. For each of the object and meta-level processes, we are developing and implementing a *Note-Assess-Guide (NAG) procedure*. The Note phase detects anomalies, the Assess phase hypothesizes what caused the anomalies, and the Guide phase performs a suitable response. This research will enable more robust behavior in autonomous intelligent systems because the capabilities lead both to recovery in the face of surprise and to more effective learning. The approach of this research as stated above has not changed from that specified in the original proposal.

During the current period, we have initiated a significant campaign to specify the means by which autonomous agents (hardware or software) can respond to unexpected events and situations in complex dynamic environments. Depending upon the particular expectation violation and context, responses can vary widely but include asking questions; seeking help; attempting to retry or repeat an action; and performing recursive problem-solving. Our focus this period has been on the last alternative. That is given an anomaly, we have developed algorithms for assessing the problem and generating a goal to solve the problem. The goal can then be passed to a standard problem-solver or planner for solution. The difficult tasks are the initial problem recognition and goal generation, not the goal achievement (i.e., plan generation and execution) itself. We have extended the MIDCA cognitive architecture in service of these directions.

Concise Accomplishments

The research team is developing an integrated theory of intelligent action, perception, cognition, and metacognition, is constructing a Metacognitive, Integrated Dual-Cycle Architecture (MIDCA) for this theory, and is applying an implementation of this architecture to scenarios in the domain of robotic mission rehearsal. During the last year, the team has been integrating existing implemented systems to form an initial architectural structure that approximates the major functions of MIDCA. These include the SHOP2 hierarchical planning system and the Meta-AQUA integrated multistrategy learning system. In later years, these pre-existing systems will be ablated as new components are tested, finished and inserted. We have also produced substantial progress on the D-track of the NAG procedure above. Last year's work on using the A-distance metric for anomaly detection has been matured, and we have collected over one billion observations used in empirical evaluation of the Note phase of the procedure. Additionally we started implementation of a D-track Assess phase implementation. This research uses a neural network to induce proto-type nodes for observed anomalies, and we are developing methods to prioritize explanations and responses (Guide phase) that have proven effective with past anomalies in proto-type categories. Finally we have been implementing a base-line performance system for the NAG procedure. This base-line uses two machine learning algorithms to create decision tree structures that generate goal-responses given observed state input. This base-line Note-Guide procedure will be compared empirically to the Note-Assess-Guide procedure to evaluate the relative performance of our algorithms. The data are encouraging and the research community has reacted favorably. Several new publications support our claims herein.

Expanded Accomplishments

The project is currently in year the second year of a three year tenure and has made significant progress. We are on schedule for spending commitments and expect to reach our expenditure target. During the last year, we have moved along our schedule quite well, integrating a set of components to comprise an initial implementation of the MIDCA architecture and adding significant detail to the Note-Assess-Guide (NAG) procedure within the architecture. We are currently working on transferring these results to the metacognitive layer and expect further results soon. Theoretically we have elaborated the structure and contents of the architecture at the meta-level. This we believe will lead to a culmination of effort in the final and third year of the project.

We have organized the subsections below by the task decomposition from the original proposal (see also Table 2 in the Work Plan section). The research tasks are as follows.

1. System Integration
2. Ontology Development
3. Domain and Scenario
4. Note-Assess-Guide Procedure
    i. Note Phase
    ii. Assess Phase
    iii. Guide Phase

5. Self-Models (Year 3)
6. Theory/Architecture
7. Evaluation

Empirical results will be presented within the subsection relating to individual processes rather than as a separate Evaluation section of its own. Task 2 is later in this report because it describes work by UMBC.

System Integration (Task 1)

The work performed during this current reporting period has resulted in the first computational implementation of the MIDCA architecture. MIDCA consists of "action-perception" cycles at both the cognitive (i.e., object) level and the metacognitive (i.e., meta-) level (see Cox, Maynord, Oates, Paisner, & Perlis, 2013). 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.

The MIDCA_1.0 model (Maynord, Cox, Paisner, & Perlis, in press) includes a complete planning-acting and perception-comprehension cycle at the cognitive level, and it incorporates a simple world simulator. The planning component integrates the SHOP2 hierarchical network planner (Nau, Au, Ilghami, Kuter, Murdock, Wu, & Yaman, 2003).[1] The comprehension component integrates the various programs developed under the project for the NAG procedure. The simulator takes actions from the planner, calculates the changes to the world, and then passes the resulting state to the comprehension component. Comprehension examines the input for anomalies and generates new goals for the planner is warranted.

Domain and Scenario (Task 3)

At the current time, we have the full plan->simulate->goal-generate->plan cycle implemented for a very simple variation of the blocks domain. This world includes blocks and pyramids. But instead of arbitrary block stacking, the purpose of plan activity is to build houses such that blocks represent the wall structure and pyramids represent house roofs. Within this domain object may catch on fire and so impede housing construction. New operators can extinguish fires and find arsonists. We have preliminary results that demonstrate better house construction performance using a top-down

[1] Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research 20*, 379–404.

goal generation strategy compared to a statistical approach. Some of these details and a description of the individual interpretation methods used for comprehension are contained below.

The current scenario implements a house building cycle that transitions through the states as shown in Figure 1. Each time the system reaches either state (a) or (c), another house is finished.

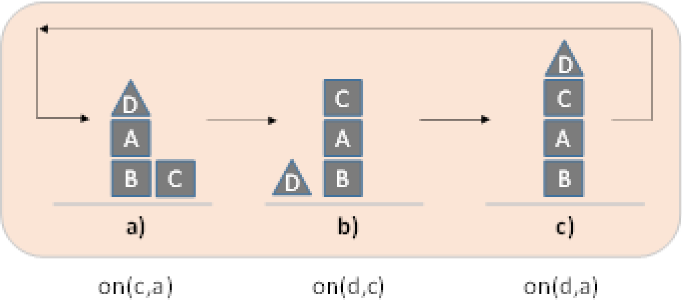

Figure 1. Housing construction cycle. The state below each panel represents the goal that transitions the system to the next panel. Houses are considered built at panels (a) and (c).

Interpretation (Task 4)

In MIDCA comprehension consists of a perceptual, an interpretive, and an evaluation component. Within this the NAG procedure is part of the interpretation process. This year has resulted in significant development of the interpretation elements, with empirical results for each of the Note, Assess, and Guide phases of the procedure. For the NAG procedure, we have developed both data-driven as well as knowledge rich approaches to cognition and metacognition. We call the former approach *the D-track* and the latter *the K-track* (Cox, Maynord, Oates, Paisner, & Perlis, 2013). Most of our progress this year has been with respect to the D-track processes (but see Cox, 2013),[2] and we anticipate greater focus on the K-track next year.

Note Phase (Task 4i)

In terms of *the Note phase* of the NAG procedure, we have developed a novel method for detecting change in symbolically represented environmental states. We apply a numerical function called the A-distance metric to streams of predicate states and look for locations in the stream that signal a change in the underlying probability distribution. For each predicate in a particular state, there is a specific number of the relations that are true. This defines a vector for each state through which a plan sequence traverses. We have been able to successfully detect anomalies at various levels of intensity across two artificial domains at this time. Figure 2 shows the F1 statistic that combines precision and recall. Here the left panel provides results from the blocksworld domain; whereas the right panel uses the logistics domain. The results vary as a function of anomaly intensity. When the observations are all anomalous (intensity 100%), the performance is accurate. With less anomalous observations (e.g., intensity 50%), performance degrades. The problem with this method is that the user must determine a realistic threshold value to use for the epsilon parameter. Various values results in a different mix of false positives and correct rejections. This efficient bottom up method adds to MIDCA's capability, and it represents a novel new way of detecting

[2] Cox, M. T. (2013). Goal-driven autonomy and question-based problem recognition. Unpublished.

change in symbolic environments. See Cox, Oates, Paisner, & Perlis (2012; 2013) for further details.

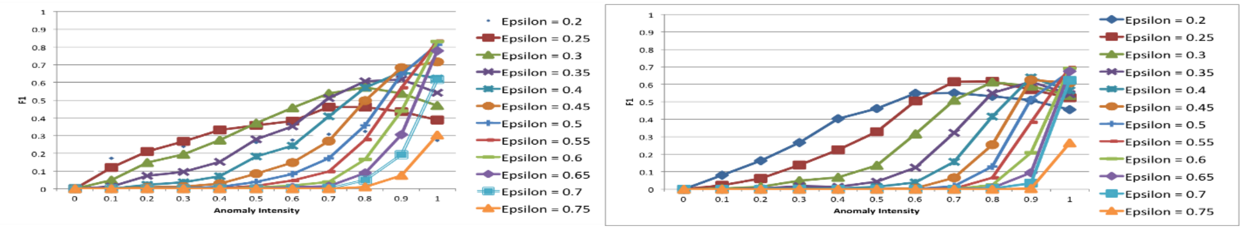

Figure 2. F1 as a function of anomaly intensity. Left graph is blocksworld results. Right graph is logistics

Assess Phase (Task 4ii)

In terms of *the Assess phase* of the NAG procedure, we have developed an interesting approach. The Note phase provides a Boolean that recognizes an anomaly in a window of time, but it provides little additional information other than the A-distance values for each predicate. We have recently implemented a Growing Neural Gas (GNG) network that takes as input the stream of A-distance values and outputs a characterization of the anomaly when present (Paisner, Perlis, & Cox, in press).[3] The information provides an anomaly type, a magnitude, and a valence for each anomaly.

Figure 3 shows a small simple example network with two normal nodes within a sphere centered at the origin and two arcs that represent different types of anomalies. The first anomaly type was created by removing the unload-truck operator from the domain model; the second anomaly type depended upon the removal of unload-airplane; and the third was a combination of both anomaly types. The arc extending along the inside-airplane predicate dimension corresponds to anomalies caused by removing the unload-airplane operator from the domain model; whereas the red arc extending along the inside-truck predicate dimension corresponds to an anomaly type caused by removing the unload-truck operator from the domain.

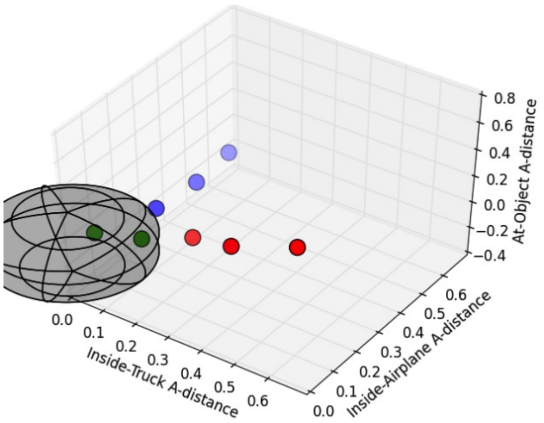

Figure 3. GNG nodes in a three dimensional space with non-anomalous nodes within a sphere centered at the origin

Interestingly this GNG D-track method also provides a way to detect anomalies without having to manually determine a good epsilon threshold. As shown in Figure 4, the performance of the

[3] See also Bhargava, et al. (2012) and Shamwell, et al. (2012) for other results using GNG networks.

composite A-distance/GNG method works nearly as well as the best epsilon value (i.e., 0.25), but it does this automatically without supervision.

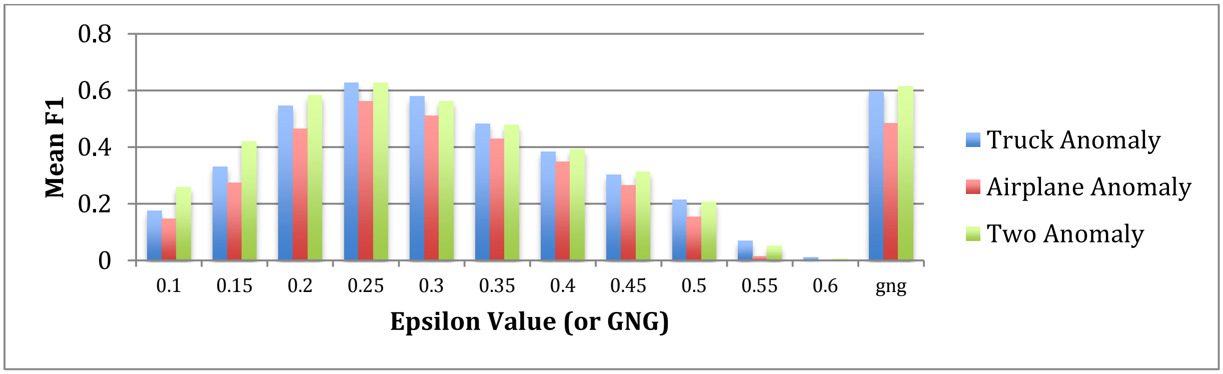

Figure 4. Mean F1 as a function of epsilon value

Guide Phase (Task 4iii)

For *the Guide phase* of the NAG procedure, we have developed two methods that we are currently comparing and contrasting. Both are concerned with generating an attainment goal when anomalies occur. The D-track method uses a combination of two machine learning algorithms to induce classifiers (Maynord, Cox, Paisner, & Perlis, in press). A combination of the Tilde algorithm and FOIL produces a goal classifier we call a TF-Tree structure. Once trained with appropriate examples, the classifier recognizes a goal expression given an observed state of the environment. The accuracy of a particular TF-Tree varies depending upon the size and constitution of the training corpus. In our domain, the variation of Table 1 is typical.

Table 1. Goal generation accuracy across training corpus sizes

Training Corpus Size	Accuracy
5	0.59
10	0.68
25	0.88
100	1.0
1000	1.0

Alternatively the K-track method explains what causes a given anomaly and generates a goal from salient antecedents of the explanation structure. Preliminary evaluation of the two methods shows that the K-track technique leads to shorter solution plans given the same anomalies (Cox, 2013). In the domain pictured in Figure 1, the K-track produces greater numbers of houses in a given time interval. The improvement is due to the fact that the K-track method generates an anticipatory goal to find and apprehend the arsonist and thus stop the fires; whereas the statistical method simply generates goals to put out fires once they are started.

Theory/Architecture (Task 6)

During the period of performance covered by this report, we have made significant result in identifying the details of the MIDCA architecture that most impact a computational approach to metacognition. For the models developed here to be realistic and cover the wide range of activities in which metacognition is involved, we must develop a detailed model of the two levels below it. That is the object or cognitive level cannot be overly simplistic if we are to have a fine level of fidelity at the meta-level. However, our claim is that the metacognitive and cognitive cycles are similar, so the effort spent at the object level will reap rewards when we finish the implementation at the meta-level. Likewise it is important to have a detailed, if not fully realistic, model of the action and perception components at the ground level. Although we are currently using a toy domain for our studies, we plan to transfer our results to a more details domain such as those already developed for the SHOP2 planner.

To appreciate the distinctions in the relationship between levels, examine the finer details of the object level as shown in Figure 5 (further details are given in Cox, Maynord, Oates, Paisner, & Perlis, 2013). 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 some *Task* that can achieve the goal. The Plan component then generates a sequence of *Actions* (π_k, e.g., an HTN plan) that instantiates that *Task* given the current model of the world (M_Ψ) and its background knowledge (e.g., semantic memory and ontologies). The plan is executed by the Act component to change the actual world (Ψ) 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.

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 GDA interpretation procedure implements the comprehension process. The procedure is to *note* whether an anomaly has occurred; *assess* potential causes of the anomaly by generating explanatory *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 (ΔM_Ψ), 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.

Likewise introspective monitoring starts with “perception” of the self (Ω) via the Monitor component. The Interpret component takes as input the resulting *Trace* (i.e.,τ_l) and the expectations in memory (π_m and g_c) to determine whether the reasoning is making sufficient progress. The Interpret procedure is to *detect* a reasoning failure; *explain* potential causes of the failure by generating explanatory *Hypotheses*; and *generate* a learning goal or attainment goal. Reasoning about the self (e.g., am I knowledgeable about the domain) and the reasoning task enables the agent to determine the difference (i.e., learning vs. attainment goal). If MIDCA produces a learning goal, the meta-level control will create and execute a learning plan to change its knowledge. Attainment goals are passed through to the object level. Given no anomaly, the Evaluate component incorporates the concepts inferred from the *Trace* thereby changing the self model (ΔM_Ω), and the cycle continues.

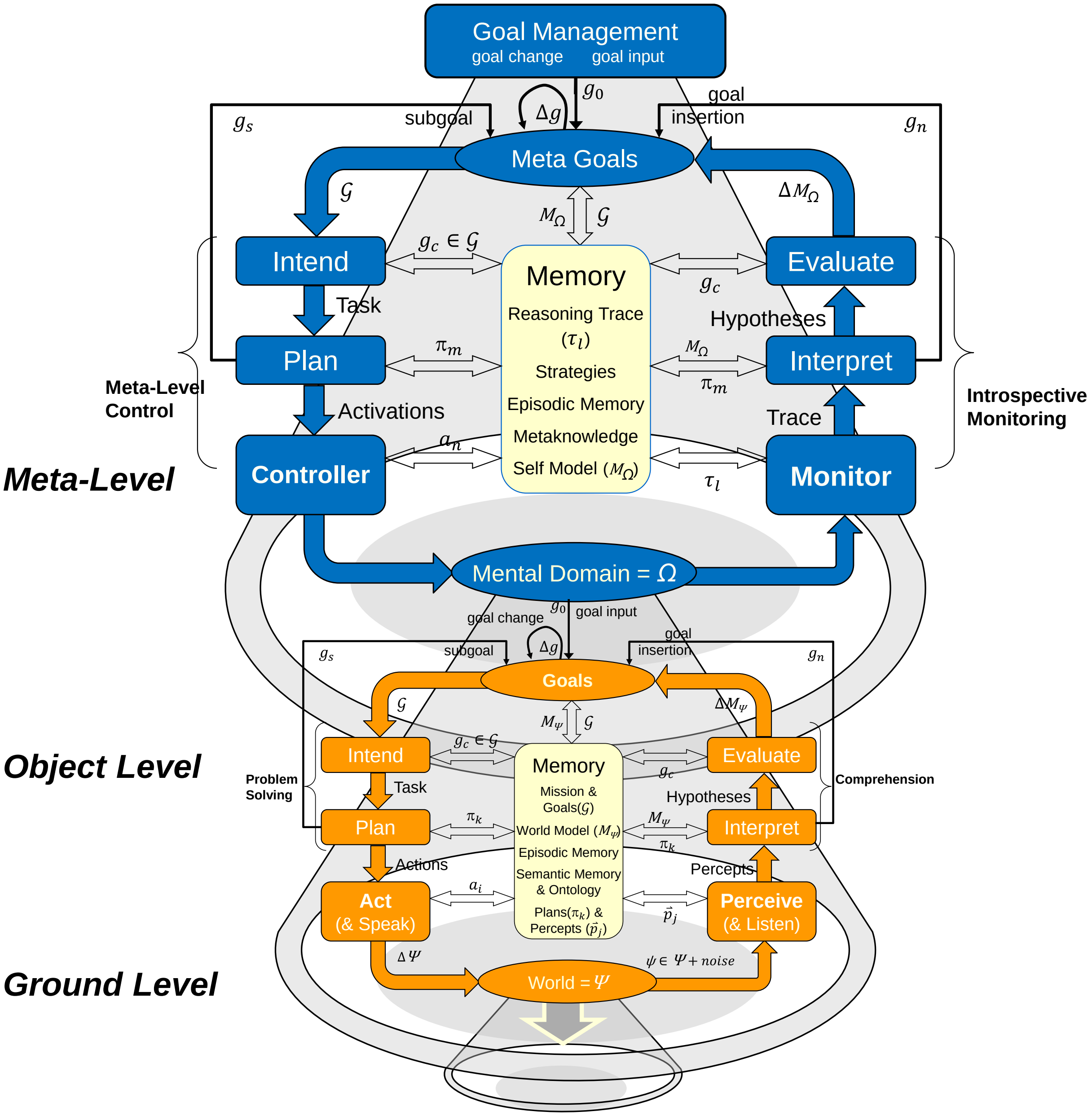

Figure 5. The MIDCA architecture

Ontology Development (Task 2)

Task 2 has changed since the proposal was written and the project started. Instead of extending the ontologies existing before the project started, UMBC has taken a different research thread as follows. During the past year, work on the project at UMBC progressed along a few different fronts. Most centrally, in terms of both time and funding, we explored methods for enabling metacognitive control over learning. Agents with learned knowledge deployed in the field need to be able to determine when the utility of that knowledge is diminished, either due to changes in the domain or the agent itself. Our general idea is to compute properties of the dynamic use of learned knowledge, determine when those properties change, and use such changes as an indicator that something is wrong. This corresponds to the Note phase of the NAG procedure.

Concretely, we looked at distributions of activations of hidden and output nodes in neural networks as the dynamic property of the use of learned knowledge. Neural networks are trained in a supervised manner but they are deployed without access to ground truth. If the distribution over inputs changes in a way that pushes the network into unknown territory, then we expect to see changes in the activation levels of hidden nodes (these levels correspond to the internal representation of the input used to produce the output). Experiments with 3-layer networks of various sizes trained with back-propagation and applied to both classification and regression problems demonstrated the utility of this approach. As one might expect, the severity of the change was correlated with our ability to detect it via activation levels, but overall the approach is rather sensitive.

This is due in part to using the A-distance metric over streaming time series to detect changes. We treat the activation level of a hidden node as an observation in a time series, and look for changes over time in this distribution. One of the drawbacks of this approach is that it is limited to uni-variate data. So we must track changes in each hidden node individually. This leads to problems with false positives. Therefore, we explores ways of adapting the basic approach to multi-variate data so that we could treat the activation of the entire hidden layer as a composite observation. We applied dimensionality reduction methods to the hidden layer activations, using both principle components analysis and independent components analysis to project the activations onto a single dimension. These approaches, however, tended to muddy the signal and make it harder to detect. A better approach was to track changes in the streams individually but aggregate them by counting the number of streams showing a change at each time step. This single stream is much more robust to false positives and thus worked quite well for detecting real changes anywhere in the projected feature space.

We also explored the use of grammar induction methods to find changes in streams of tokens. The well-known SEQUITUR algorithm processes streams of tokens in linear time and learns context-free grammars. Our idea here was to generate sequences from one grammar and then detect changes when the underlying generator is changed. The approach is to learn a SEQUITUR grammar on the sequence and monitor the rate of growth of the grammar. Typically, the grammar grows quickly as it finds structure in the sequence, but then the growth levels off as it is able to compress new parts of the sequence that look like parts it has already seen. When the generator changes, this growth rate jumps. Again, we applied the A-distance metric to increases in grammar size over time and were able to detect relatively subtle changes in the generating structure. This approach is highly scalable due to the efficiency of the underlying algorithms (SEQUITUR and the A-distance).

Finally, a student at UMBC has begun to explore what we believe to be a meta-cognitive approach to dealing with language growth, specifically to the discovery of new properties of objects as manifest through adjectives. The fundamental question is how a language learner might discover which adjectives belong on the same scale (e.g., temperature), how they are related to one another on the same scale (e.g., frigid is closer to cold than to hot), and when adjectives are learned for which no existing scale fits. This last item involves positing a new property of objects and attempting to identify related adjectives and order them on an appropriate scale. We spent significant time getting up to speed on the related literature, of which there is precious little with a computational bent, gathering examples, and using mechanical Turk to get human judgments. Our goal is to understand how people reason about relatedness of adjectives, especially as it relates to detecting the need to extend their inventory of scales.

Work Plan

Table 2 enumerates the 7 major tasks proposed for this project. We are on schedule with respect to this outline with two exceptions. First, as indicated by the accomplishments section above, we have already started to make progress on Task 4iii. Second, task 2 has changed from what we originally proposed. UMBC has pursued an investigation into metacognitive control over learning. This has complemented the research performed at Maryland very well.

During the third year of the MIDCA project, we intend to transfer the research on the object-level NAG procedure to the meta-level. This effort represents the heart of the project, and we will make stringent efforts to stay on track. Finally we will pursue Task 5, the effort to develop computational self-models. Note however that we have already made some progress by distinguishing static models of self from dynamic "processual" models (see Brody, Cox, & Perlis, 2013).

Table 2. Project Schedule and Task Assignments

<table>
<tr><th><em>Tasks</em></th><th><em>Year 1</em></th><th><em>Year 2</em></th><th><em>Year 3</em></th></tr>
<tr><td>1. System Integration</td><td colspan="3">Student + Cox</td></tr>
<tr><td>2. Ontology Development</td><td colspan="2">UMBC</td><td></td></tr>
<tr><td>3. Domain and Scenario</td><td colspan="2">Cox + UMBC</td><td></td></tr>
<tr><td>4. i. NAG Note Phase</td><td>Cox + Student + UMBC</td><td></td><td></td></tr>
<tr><td>4. ii. NAG Assess Phase</td><td></td><td colspan="2">Cox + Student</td></tr>
<tr><td>4. iii. NAG Guide Phase</td><td></td><td></td><td>Cox + Student</td></tr>
<tr><td>5. Self-Models</td><td></td><td></td><td>Cox</td></tr>
<tr><td>6. Theory/Architecture</td><td colspan="3">Cox</td></tr>
<tr><td>7. Evaluation</td><td colspan="3">Cox + UMBC</td></tr>
</table>

Major Problems/Issues

None.

Technology Transfer

Currently none. However we have been awarded a new ONR grant that will apply the results from this project to the guidance of an actual unmanned underwater vehicle.

Foreign Collaborations and Supported Foreign Nationals

None.

Productivity

Refereed Journal Articles

1. 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.

Workshops and Conferences

1. Bhargava, P., Cox, M. T., Oates, T., Oh, U., Paisner, M., Perlis, D., & Shamwell, J. (2012). The robot baby and massive metacognition: Future vision. In *Proceedings of the IEEE Conference on Development and Learning - Epigenetic Robotics 2012 (ICDL/EpiRob)*. Los Alamitos, CA: IEEE.
2. Brody, J., Cox, M. T., & Perlis, D. (2013). The processual self as cognitive unifier. In *Proceedings of the Annual Meeting of the International Association for Computing and Philosophy*. IACAP-2013.
3. Cox, M. T., Maynord, M., Oates, T., Paisner, M., & Perlis, D. (2013). The integration of cognitive and metacognitive processes with data-driven and knowledge-rich structures. In *Proceedings of the Annual Meeting of the International Association for Computing and Philosophy*. IACAP-2013.
4. Cox, M. T., Oates, T., Paisner, M., & Perlis, D. (2013). Detecting change in diverse symbolic worlds. In L. Correia, L. P. Reis, L. M. Gomes, H. Guerra, & P. Cardoso (Eds.), *Advances in Artificial Intelligence, 16th Portuguese Conference on Artificial Intelligence* (pp. 179-190). University of the Azores, Portugal: CMATI.
5. Du, T., Cox, M. T., Perlis, D., Shamwell, J. & Oates, T. (in press). From robots to reinforcement learning. To appear in *Proceedings of the 25th IEEE International*

Conference on Tools with Artificial Intelligence. Los Alamitos, CA: IEEE Computer Society.

6. Gandhi, S., & Oates, T. (2013). *Noticing changes in the utility of learned knowledge as detecting changes in streaming data*. Submitted.
7. Maynord, M., Cox, M. T., Paisner, M., & Perlis, D. (in press). Data-driven goal generation for integrated cognitive systems. To appear in C. Lebiere & P. S. Rosenbloom (Eds.*), Integrated Cognition: Papers from the AAAI Fall Symposium*. Menlo Park, CA: AAAI Press.
8. McNany, E., Josyula, D., Cox, M. T., Paisner, M., & Perlis, D. (2013). Metacognitive guidance in a dialog agent. In S. Shimizu & T. Bosomaier (Eds.), *Proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications* (pp. 137-140). Red Hook, NY: Curran Assoc.
9. Paisner, M., Perlis, D., & Cox, M. T. (in press). Symbolic anomaly detection and assessment using growing neural gas. To appear in *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence*. Los Alamitos, CA: IEEE Computer Society.
10. 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.

Award Participants

Faculty

- **Michael T. Cox** (University of Maryland, Institute for Advanced Computer Studies)
- **Tim Oates** (University of Maryland, Baltimore County, Department of Computer Science and Electrical Engineering)

Students

- **Sunil Gandhi** (Ph.D. student, UMBC, Department of Computer Science and Electrical Engineering)
- **Michael Maynord** (Ph.D. student, University of MD, Department of Computer Science)
- **Elizabeth McNany** (Ph.D. student, University of MD, Department of Computer Science)
- **Matthew Paisner** (Ph.D. student, University of MD, Department of Computer Science)