OVERCOMING THE PROBLEM OF BRITTLENESS WITH THE METACOGNITIVE LOOP

FINAL REPORT

Principal Investigators:

Donald Perlis (PI)
University of Maryland - College Park
College Park, MD 20742

Michael Anderson (Co-PI)
Franklin & Marshall College
Lancaster, PA 17604

Tim Oates (Co-PI)
University of Maryland - Baltimore County
Baltimore, MD 21250

Grant Number: FA9550-06-1-0091

ABSTRACT:

The metacognitive loop (MCL) is an architecture for automated noting, repairing, and learning from errors. Initial work on this award involved builing special-purpose MCL programs for each individual application domain. Later in the award period a more uniform approach was designed that employs a general framework providing ontologies for types of Indications, Failures, and Repairs. This allows the past results in different domains to be achievable by a single MCL module, changing only the domain and the IFR ontologies. As a consequence, the investigators are now positioned (starting in 2009, with a new award) to begin to build a general-purpose MCL module that, when "attached" to a given host program H and an initial set of IFR ontologies, can adapt to the domain that H lives in (and in the process adapt the ontologies to better fit that domain) so that H+MCL guides itself to become less brittle.

OBJECTIVES:

The objectives of this project were to design, build, and test in a variety of domains a new architecture for dealing with the brittleness problem, i.e., the appearance of anomalies during execution due to unforeseen circumstances. Specifically, the meta-cognitive loop (MCL) was to be employed for this purpose, based on the three processes of noting an anomaly, assessing it, and guiding a response into effect.

ACCOMPLISHMENTS:

1. We implemented a pilot MCL project consisting of an MCL enhanced AI player of the tank game called Bolo. The player can perform the basic tasks of "searching" for pillboxes and "rescuing" them. This work was reported in publication (8). Our major conclusion from this pilot experiment was that MCL-enhancements can greatly benefit systems such as the Bolo brain. Some of the implemented features of the MCL-enhanced brain are as follows:

   (a) The brain maintains operator models for different Bolo actions. Each operator model is characterized by the preconditions and effects of the action that it specifies. The MCL-enhanced brain can adapt operator models based on experience, learning new preconditions and effects.

   (b) The brain maintains its plan of actions as a hierarchical task network.

   (c) The MCL-ehanced brain creates expectations for each action it initiates using the effects specified in the operator model of that action. It then monitor expectations to determine successful completion.

   (d) The MCL-enhanced brain has a "note" module that notes expectation violations, and "assess" module that classifies the violations and attempts to find a cause, and a "guide" module that enacts a response. Responses can include: replanning the hierarchical task network, employing a Means-End-Analysis, and refining the operator model. It can also decide to do nothing.

(e) The MCL-enhanced brain improved performance considerably (measured by how long the tank is kept alive). This result, plus the result of other previous pilot studies that included a natural language interface and a reenforcement learner, we redesigned and built a stand alone MCL unit, discussed below.

2. Based on our experience with several pilot implementations of the Meta-Cognitive Loop (MCL), we designed and built a stand alone MCL unit. This unit is meant to be a system that "sits on top" of other systems that need monitoring. That is, instead of building MCL reasoning into a system, we aimed to build a system that, with a small amount of interfacing, could monitor any other intelligent system. The new MCL currently in the build and test phase, and now has the following properties:

   (a) The new MCL is made up of three ontologies that correspond to the three phases "note", "assess", and "guide", that we had originally hypothesized to be an accurate way of describing how humans handle anomalies. The three ontologies are now referred to as the "indications", "failures", and "responses" ontologies. The indications ontology represents the possible indicators to an anomaly, such as a sensor reading that is off mark or a missing piece of communication. The failures ontology encodes all of the (what we believe to be finite) ways in which a system might fail, such as a sensor malfunction or an incorrect world model. The responses ontology provides the ways in which a system might respond to any type of anomaly, from recalibrating a sensor to simply ignoring the problem.

   (b) Each of the ontologies is made up of a hierarchically organized set of nodes with each node connected to each other node. Each node corresponds to what we believe to be the essential categories of possible indications, failures, and responses. The connections between nodes correspond to the relationships between the classifications. For instance, in the indications ontology, there is a node labelled "sensor-reading-greater-than-expected" and another "sensor-reading-less-than-expected", which are dominated by the node "sensor-reading-not-as-expected". There are many

such nodes in each ontology that are not just connected within ontologies, but also across ontologies.

(c) The host system (i.e., the system that MCL is monitoring) is connected to MCL via "fringe" nodes in the indications and responses ontologies. The fringe nodes in the indications ontology represent host-specific expectation violations, while the fringe nodes in the responses ontology represent host-specific actions.

(d) The ontologies use a Baysian method of propogating information from node-to-node and across ontologies. When an expectation is violated, that information is fed to MCL via the fringe nodes. The information is propagated up the indications ontology, making an increasingly abstract classification of the indication. The information is passed over to the failures ontology, and then to the responses ontology. Each node in the propagation measures distinct rates of activation, ultimately providing a unique portrait of the anomaly. Based on that portrait, MCL chooses a response and sends it to the host system.

(e) The new architecture allows MCL to learn associations between ontologies and classifications, and hence, it can adjust its advice given experience.

(f) The new architecture aims to generalize MCL in that, after we have trained it on several different kinds of systems, it should work on many other systems it has not had experience with. Our new testbed was designed to implement such a training and test the results.

3. One of our major MCL testbed systems is a natural language agent called ALFRED. ALFRED was used in several pilot experiments with MCL and these results were instrumental in convincing us of MCL's power. MCL, we believe, would truly be generalizable if it works in the realm of natural language. In fact, it may be a key missing component to current natural language systems. Thus, we decided to redesign ALFRED from top to bottom in order to provide a more solid base for dialog correction and repair. The main impetus for this change was that for ALFRED to notice anomalies in dialog and react to them, it must be capable of reasoning about its language skills and any dialog

that it engages in. This requires that ALFRED not dumbly apply its language skills, but for its language knowledge to be in the same representational schema and be just as accessible as any other kind of knowledge. ALFRED currently has the following properties:

(a) ALFRED has an English lexicon consisting of some names, nouns, verbs, prepositions, and articles. Each lexical entry records properties of the word including multiple forms, (multiple) spelling, part of speech, argument structure (for verbs), etc.

(b) ALFRED has a collection of English syntax rules which are applied to user utterances to produce constituent structures.

(c) We have begun implementing a semantic component that takes the constituent structures of utterances and produces a logical form. These logical forms include the propositional content of the utterance as well as any speech act information.

(d) Alfred has a non-linguist concept space for representing concepts and their relationships. Lexical entries are related to these non-lingusitic concepts by a simple predicate *label for, so that the word 'move' is a* label for the concept of moving. Logical forms of utterances are represented in terms of these non-linguistic concepts.

(e) ALFRED has knowledge of the command language(s) used to communicate to the domains that it controls. In our current testbed, ALFRED will be connected to a virtual Mars domain consisting of several virtual robot rovers. These rovers can accept commands, perform actions, report sensor readings, etc. ALFRED acts as the interpreter of English into "Roverese" and back into English. Therefore, we have designed ALFRED so that it has knowledge of Roverese that mirrors its knowledge of English, i.e., it has a Roverese lexicon, syntax rules, and semantic rules. The Roverese lexicon is associated with non-linguistic concepts in the same way that English is.

(f) ALFRED is connected to a Mars rover simulation in which one or more virtual rovers accept commands delivered by ALFRED and report statistics to ALFRED.

4. We built a virtual Mars rover domain as a testbed for both ALFRED and MCL. The Mars domain currently has the following properties:

   (a) The domain is capable of hosting more than one rover at a time.

   (b) There is a map of locations which each rover can move around in.

   (c) There is a command language for communicating with the rover(s) used both for the issuance of commands directed at the rover and the reporting of sensor data by the rover to the user.

   (d) We have begun creating the fringe nodes to hook up an MCL unit to the rovers.

5. We built a new reasoning engine called ALMA 2.0 for specifying the contents of ALFRED's knowledge base. Like the first ALMA, ALMA 2.0 allows for controlled reasoning (through time steps) in the presence of contradictions.

6. The following PhD dissertation was completed:

   Waiyian Chong, "Reflective Reasoning." 2006, University of Maryland.

   ABSTRACT: This dissertation studies the role of reflection in intelligent autonomous systems. A reflective system is one that has an internal representation of itself as part of the system, so that it can introspect and make controlled and deliberated changes to itself. It is postulated that a reflective capability is essential for a system to expect the unexpected—to adapt to situations not forseen by the designer of the system. Two principal goals motivated this work: to explore the power of reflection (1) in a practical setting, and (2) as a method for approaching bounded optimal rationality via learning. Toward the first goal, a formal model of reflective agent is proposed, based on the Beliefs, Desires and Intentions (BDI) architecture, but free from the logical omniscience problem. This model is reflective in the sense that aspects of its formal description, comprised of set of logical sentences, will form part of its belief component, and hence be available for reasoning and manipulation. As a practical application, this model is suggested as a foundation for the construction of conversational agents capable of meta-conversation, i.e., agents that can reflect on the ongoing conversation. Toward the second goal, a new reflective form of reinforcement

learning is introduced and shown to have a number of advantages over existing methods. The main contributions of this thesis consist of the following: In Part II, Chapter 2, the outline of a formal model of reflection based on the BDI agent model; in Chapter 3, preliminary design and implementation of a conversational agent based on this model; in Part III, Chapter 4, design and implementation of a novel benchmark problem which arguably captures all the essential and challenging features of an uncertain, dynamic, time sensitive environment, and setting the stage for clarification of the relationship between bounded-optimal rationality and computational reflection under the universal environment as defined by Solomonoff's universal prior; in Chapter 5, design and implementation of a computational-reflection inspired reinforcement learning algorithm that can successfully handle POMDPs and non-stationary environments, and studies of the comparative performances of RRL and some existing algorithms.

PERSONNEL SUPPORTED

Donald Perlis (PI)
Michael Anderson (co-PI)
Tim Oates (co-PI)
Darsana Josyula (post-doc)
Matt Schmill (post-doc)
Scott Fults (post-doc)
Waiyian Chong (GRA)
Shomir Wilson (GRA)
Hamid Haidarian Shari (GRA)
Percy Tiglao (hourly paid student)
Sarit Kraus (visiting faculty)

PUBLICATIONS

1. A review of recent research in reasoning and metareasoning. Michael L. Anderson, Tim Oates. AI Magazine, 28(1): 7-16, 2007. (Article featured on the cover of AI Magazine)

2. A self-help guide for autonomous systems. Michael L. Anderson, Scott Fults, Darsana P. Josyula, Tim Oates, Don Perlis, Matt D. Schmill, Shomir Wilson and Dean Wright. AI Magazine, 29(2): 67-76, 2008.

3. Active logic semantics for a single agent in a static world. Michael L. Anderson, Walid Gomaa, John Grant and Don Perlis. Artificial Intelligence 172: 1045-63, 2008.

4. Ontologies for Reasoning about Failures in AI Systems. Matt D. Schmill, Darsana P. Josyula, Michael L. Anderson, Tim Oates, and Scott Fults. Ontologies for reasoning about failures in AI systems. Proceedings of the First International Workshop on Metareasoning in Agent-Based Systems, 2007.

5. ReGiKAT: (Meta-)Reason-guided knowledge acquisition and transfer – or Why deep blue can't play checkers, and why today's smart systems aren't smart. Michael Anderson, Tim Oates, and Donald Perlis. Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Paris, 2006.

6. Toward Domain-Neutral Human-Level Metacognition. Michael L. Anderson, Matthew D. Schmill, Tim Oates, Darsana Josyula, Dean Wright, Scott Fults and Shomir Wilson. Conference proceedings of the 8th International Symposium on Logical Formalizations of Commonsense Reasoning, Hawaii 2007.

7. The metacognitive loop I: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. Michael L. Anderson, Tim Oates, Waiyian Chong and Don Perlis. Journal of Experimental and Theoretical Artificial Intelligence, 18(3): 387-411, 2006.

INTERACTIONS/TRANSITIONS

Patents: none
Inventions: none
Awards: none

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
	FINAL REPORT	1 Feb 06 – 30 Nov 08

4. TITLE AND SUBTITLE
OVERCOMING THE PROBLEM OF BRITTLENESS WITH THE METACOGNITIVE LOOP

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-06-1-0091

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
DR DONALD PERLIS

5d. PROJECT NUMBER
2311

5e. TASK NUMBER
FX

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF MARYLAND, COLLEGE PARK
INSTITUTE FOR ADVANCED COMPUTER STUDIES
A.V. WILLIAMS BUILDING
COLLEGE PARK, MD 20742

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR.RSL
875 NORTH RANDOLPH ST, ROOM 3112
ARLINGTON, VA 22203-1678

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VH-TR-2012-0062

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION
APPROVE FOR PUBLIC RELEASE:DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The metacognitive loop (MCL) is architecture for automated noting, repairing, and learning from errors. Initial work on this award involved building special-purpose MCL programs for each individual application domain. Later in the award period a more uniform approach was designed that employs a general framework providing ontologies for types of Indications, Failures, and Repairs. This allows the past results in different domains to be achievable by a single MCL module, changing only the domain and the IFR ontologies. As a consequence, the investigators are now positioned (starting in 2009, with a new award) to begin to build a general-purpose MCL module that, when "attached" to a given host program H and an initial set of IFR ontologies, can adapt to the domain that H lives in (and in the process adapt the ontologies to better fit that domain) so that H+MCL guides itself to become less brittle..

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*