

---

## A Broad Vision for Intelligent Behavior: Perpetual Real-World Cognitive Agents

---

**Don Perlis**

PERLIS@CS.UMD.EDU

**Michael T. Cox**

MCOX@CS.UMD.EDU

Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA

**Michael Maynard**

MAYNORD@UMD.EDU

**Elizabeth McNany**

BETH@CS.UMD.EDU

**Matthew Paisner**

PAISNER@CS.UMD.EDU

**Vikas Shivashankar**

SVIKAS@CS.UMD.EDU

Computer Science Department, University of Maryland, College Park, MD 20742 USA

**Jared Shamwell**

EJSHAM@UMD.EDU

Neuroscience and Cognitive Science Program, University of Maryland, College Park, MD 20742 USA

**Tim Oates**

OATES@CS.UMBC.EDU

Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250 USA

**Tongchun Du**

TONGCHUNDU@GMAIL.COM

Department of Automation, Harbin Engineering University, Harbin, China

**Darsana Josyula**

DARSANA@CS.UMD.EDU

Computer Science Department, Bowie State University, Bowie, MD 20715 USA

**Manuel Caro**

MFCAROP@UNAL.EDU.CO

Department of Educational Informatics, University of Cordoba, Monteria, Colombia

### Abstract

We describe ongoing work toward automating human-level behavior that pulls together much of traditional artificial intelligence in a real-time robotic setting. Natural-language dialog, planning, perception, locomotion, commonsense reasoning, memory, and learning all have key roles in this; and metareasoning is a sort of glue to guide the robot through rough spots.

### 1. Introduction

We are in the middle of a multi-year research program aimed at pulling together many parts of artificial intelligence in a suitable manner so that an agent constructed along such lines may come

closer to “human-level” performance. The target performance spans not only a wide range of specific tasks, but also includes the ability to learn about, adapt to, and adjust both its environment and its own abilities (not unlike a human baby). While strict mimicking of human behavior is not our goal, we are mindful that we have much to learn from human behavior, and do not hesitate to make use of both established and intuitive insights therein. See Shamwell et al. (2012) for more along these lines.

Thus learning/adaptation occupies a special place in our vision. But so do general reasoning, perception, and action. Indeed, one theme that is central to our effort is that a system can and should be able to learn about the effects of its own actions via a combination of perception, inference and planning. Not only that, but a particular form of inference – metareasoning – is critical for us. For instance, such an agent may decide (by reasoning about its past episodes of reasoning) that it is not very good at solving certain kinds of problems and may then ask for help in learning to deal with such a problem. That in turn may involve natural language processing, and so on.

This paper is organized as follows: We will begin with a running example to illustrate much of our ideas; then we will focus separately on some of the pieces that the example depends on, where we already have had some successes; we then give a brief review of related work; and finally we will describe our current and future plans. The majority of this paper is devoted to the elements that we believe are essential to autonomous agents that must operate in the real world. Both reasoning and meta-reasoning fall into this category, with our approach to the latter involving a time-situated logic called active logic and a lightweight, general purpose architecture for meta-reasoning called the Meta-Cognitive Loop. Because we want our agents to engage with humans, we describe our experience with a dialog system based on active logic called ALFRED. A discussion of reinforcement learning follows that, given the prominent role that it plays in (low-level) learning to interact with the world. Next we explore an often overlooked aspect of autonomy - goal generation - and discuss it in the context of an overall cognitive architecture called MIDCA, and finish with a discussion of how memory interacts with all of these components.

We offer a cautionary note: while our main example involves a robot, we are not here suggesting a project in traditional robotics at all. Rather our aim is along the lines of the original conception of AI as the computational study of intelligent human-level behavior. Thus swarm robotics, multi-agent systems, and the like, while important in their own right, have little to do with what we are investigating here. The science-fiction image of a household robot that can do many things while learning on the job is closer to our vision; and to those who say this is not where big successes are to be found, we reply: that is the issue, and we are offering to test it in what we think is a new way. Indeed, if this cannot be done, then it would seem that human-level intelligent behavior is not largely computational after all, and that in itself would be big news.

## 2. Running Example

*Consider a robot – Robbie – whom we have assigned the task of obtaining a particular book and bringing it to us. We have told Robbie to look for the book in room 128, and that we need the book before noon. It is now 11:30am. Robbie sets off for room 128, having previously learned a floor plan that she now consults to plan a path. She also marks the task details so that they*

*remain in working memory; she knows from past experience that without this precaution she can lose sight of details and it can take lots of time to recover them, and that in this case time matters.*

We pause here to give some background about Robbie. She is one year old, having had that much near-continuous training since being first turned on (and having stored her acquired knowledge appropriately in various forms of memory). That is, Robbie is a perpetual agent: she has a lifetime of her own and is not simply turned on briefly when we want her to do something. At first, she knew very little – just what the factory had installed in her KB – and even less about how to perform physical actions with any dexterity. But she has an internal goal of sorts, based on the maxim that “knowledge is good.” So when she is not working on an assigned (exogenous) task she pursues the endogenous goal of trying to learn about whatever she can. Of course, that can quickly lead to disaster since some explorations can lead to injury to herself and others; so she also has some initial cost-benefit information which she augments and modifies as she learns from experience including human interaction. In addition, she has had to learn to “see” – that is, to interpret her perceptual data-flow – and to relate that to her own activity: if she moves forward, her visual flow changes in one way; if she rotates, it changes in another; and if she reaches forward she sees her own arm, etc. She also has had to learn what books are: how they look, that they can be picked up, and so on. All this has taken up much of her first year.

To render the present context a bit more fully: we are studying for an exam that will take place at 12:30pm that same day; the book might possibly be helpful but we are fully occupied in memorizing some key items and cannot take the time to get the book ourselves; and we must set out for the exam by noon. Now we return to our story: Robbie is looking for a book...

*Along the way, Robbie encounters a cluttered 10m section of hall, which slows her down considerably. After some time spent in trying to pass through that section, she decides instead to go on another, longer, route that, she anticipates, is not cluttered.*

*On arriving at the intended room, Robbie expects to find that it has the number 128 but instead observes 123. She puzzles about this and checks the doors on both sides, which are numbered 127 and 129. She looks again at the 123 and supposes that it either is a mistake or that the 8 has somehow degraded and now looks like a 3, and that in any case she is indeed at the correct room. She looks at the suspect 3 more carefully, detects what might have been the degraded part of an 8, and makes note of this for future use in reading numbers.*

*But the door to the room is closed. Robbie tries the knob, and cannot manage to turn it, having been trained only on door handles, not knobs. She makes numerous attempts in different ways, using one or more of her problem-solving algorithms, but still fails.*

*Robbie decides she needs help, and comes back to us for advice. It is now 11:45. We tell her that we did not realize that the door to room 128 was closed, and that knobs are too hard for her grippers to manage, but if she uses a key, the door will open without having to turn the knob at all. We give her a key, she starts off, and arrives again at room 128. After some effort the key allows entry, and as a result, Robbie learns a new method for opening the door.*

*However once inside she discovers that the book is not at the shelf position where it should be. It must have been misfiled, and there are hundreds of books to look through (the room seems to be a library of sorts). So she realizes that with all the time already taken – it is now 11:55 – it is very unlikely she will be able to find the book and still get it to us by noon. She returns and tells us this. We agree and tell her not to bother after all, and thank her for her efforts.*

A great many things happen in this example. Yet they can be broken down into pieces most of which, separately, have been the subjects of enormous amounts of (highly successful) research. On the other hand, putting it all together effectively is far more than merely assembling the pieces. New issues arise – or become far more important – in the aggregate; for example, the need to know (and reason about) one’s own knowledge and behavior; the need to keep track of ongoing time; the need to mix plans and speech and others’ interests all in the same bit of reasoning; the need to recognize when events are anomalous; the need to notice salient features of events for future use; and the need to manage memory efficiently so as to keep relevant information in working memory and irrelevant information out.

This example may seem absurdly difficult, far beyond what anyone has any hope of achieving in the next several decades. But we think not. In the following several sections, we will describe work we have done on a number of topics closely related to the above kind of scenario. We start with the need to reason about one’s own knowledge in the face of contradictory information, and to monitor time-passage during reasoning.

### **3. Reasoning and Metareasoning: Active Logic**

Active logics (Anderson, Josyula, Okamoto, & Perlis, 2002) are a family of formalisms that combine inference rules with a constantly evolving measure of time (a “now”) that itself can be referenced in those rules. At each time step, all possible one-step inferences are drawn by applying inference rules once to the present (working memory) knowledge, and marked with a timestamp. Allowing inferences only to be made based on one-step inferences on present knowledge, and not made by applying inference rules iteratively until the next time step, helps mitigate the “omniscience problem”, where all implications are treated as derived at once. By explicitly situating reasoning in time this way, contradictions can be dealt with as they arise in the inference process. In our example, Robbie encounters various contradictions, such as believing the room number is 128 but seeing it to be 123.

Time steps also aid in reasoning about past reasoning, and in the derivation of future theorems. Robbie for example would use the present value of “Now” in determining if there is sufficient time for her to complete her task (see Brody, Cox, & Perlis, 2013 for some details). Active logic differs from other temporal logics which lack a “now” represented as a changing time value; these other logics simply discriminate between a fixed past, present, and future.

Active logic is a non-monotonic reasoning scheme, meaning that inferences made in the past can be rejected and replaced with better ones in the present. The tagging of the reasoning process with time stamps allows the use of a belief’s history of acceptance/rejection during present reasoning; when direct contradictions arise in the knowledge base, this information can be of use. In particular, conflicts between expectation and observation can be recognized and reasoned about.

Each active logic belief is tagged with a unique identifier; this allows the reasoning mechanism to refer to inferences or assign properties to them – for instance, a belief can be distrusted, removed or assigned/reassigned a higher/lower priority. Thus, active logic is a natural mechanism for default reasoning (Purang, 2001) as well as resource-bounded reasoning and meta-reasoning (Josyula & M’Bale, 2013).

#### 4. The Metacognitive Loop (MCL)

In our example, Robbie experienced several unexpected problems in the course of carrying out her task. To continue on mission, she needed to not only identify that there was a problem, but also implement a suitable response. We call such a reasoned anomaly-handling capability *generalized metacognition*. Ideally such a process can be largely domain-independent, involve only a modest amount of background knowledge and computation, and be implemented for any automated system. Much of our recent work has been aimed at testing this idea (e.g., see Schmill, Anderson, Fults, Josyula, Oates, Perlis, Haidarian, & Wilson, 2011). It essentially consists of three steps (the NAG-cycle): (i) monitor expectations to *note* any anomaly that might arise, (ii) *assess* it in terms of available responses, and (iii) *guide* any chosen response into place (and monitor the progress of that response for further anomalies). This requires, of course, expectations as to how things “ought” to be in the system, responses that could apply across the board to almost any type of anomaly encountered, and the ability to re-configure expectations in light of how things go. We refer to an algorithmic version of the NAG-cycle as the Metacognitive Loop (MCL) (Anderson & Perlis, 2005).

Our current generalized MCL module implements three special sets of abstract ontologies: an indications ontology for anomaly types, a failures ontology for assessment, and a responses ontology for repairs. The core of each ontology is currently implemented as a Bayesian network. These core nodes represent abstract and domain-general concepts of anomalies and how to respond to them. These nodes are linked within each ontology to express relationships between the concepts they represent. They are also linked between ontologies, allowing MCL to employ a number of Bayesian algorithms for reasoning over ontologies. Attached to the indications and responses ontologies are concrete “fringe” nodes. The fringe nodes for the indications core represent concrete, specific information about a possible expectation violation, and those for the responses core represent specific correction information. The host provides updates (e.g., sensor data) to the expectations fringe, and receives suggestions for repairs via the corrections fringe. When the expectation fringe nodes receive an update from the host, if the observed values in the update are different from the expected value specified in the fringe node, then an expectation violation occurs. The expectation violation triggers nodes in the indication ontology that correspond to the violation; the node activations propagate from the lower level, more specific fringe nodes to the higher level, more abstract indications of failures. The activations also get propagated to the failure ontology through the indications-failures inter-ontology links and to the response ontology through the failures-responses inter-ontology links. As the node activations propagate down the responses ontology to the more specific correction fringes, the correction fringe with the highest utility sends a specific correction to the host to act on.

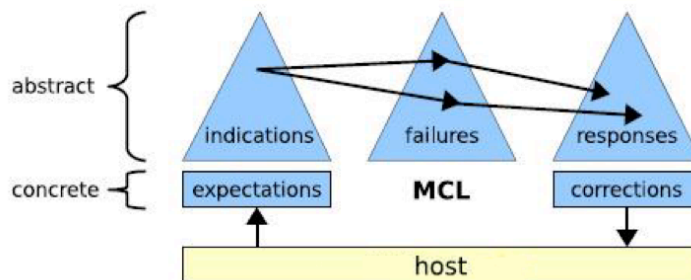


Figure 1. Metacognitive loop (MCL)

Figure 1 depicts the host (shaded in yellow), and the generalized metacognition module MCL (shaded in blue). During operation, the host can also adjust or specify new expectations based on its ongoing experience. At the input interface, expectations are directly linked to the indications ontology through indication fringe nodes. At the output interface, the responses ontology's fringe nodes are linked to a set of possible corrections that the host could employ. When an actual perturbation occurs in the host, MCL will detect the expectation violation through the input fringe nodes. It will then attempt to map it into the MCL core so that it may reason about it abstractly. MCL's reasoning process then produces an output, which is articulated through the output fringe nodes in the form of an action that the host is able to carry out.

Returning to the example, Robbie encounters her first unexpected challenge as she enters the 10m stretch of cluttered hallway. She is under a time constraint to return the book. A cluttered hallway will force her to move slowly and may result in task failure. MCL can handle this situation in a number of ways. Robbie may have begun with outdated information and could not have realized this hallway was cluttered before reaching it; perhaps she would have chosen a different path had she known. Realizing the clutter would slow her down but also having already traveled to this stretch of hallway, Robbie estimates how long it should take her to move through the hallway and compares it with estimates for alternate routes. From her current location, travel via the cluttered hallway, with time added for the clutter, still presents the shortest estimated time to her goal and she decides to continue as planned. However, Robbie soon realizes that she is taking far longer than expected to move through the cluttered hallway. Sensing another expectation violation and reasoning about its cause and possible responses, Robbie decides to take another route. Alternatively, Robbie may not have a priori understood the negative relationship between clutter and travel time and started only with an expectation of the amount of time needed to move through a 10m stretch of hallway. When this expectation was violated, Robbie notes that something is wrong and reasons that there is something wrong with her current route, resolves to find a new route, and notes the relationship between clutter and travel time.

Continuing with the example, when Robbie reaches what she believes to be her destination, she may reasonably expect the room number written on the door to match the room number of her destination. When she reads '123' instead of '128', MCL would note an expectation violation and begin initiating behaviors to resolve the violation (in this case gathering further information by checking the two adjacent doors). In facing her third obstacle, Robbie notes a difficulty in opening the door and after initiating several behaviors aimed at resolving the problem, eventually concludes that she does not have the necessary abilities to achieve her goal and asks a human handler for help. Similarly, after using the key to open the door, Robbie realizes that the anticipated time to find the book will result in completion after the specified deadline. Having no known ability that would allow her to complete the task any faster, she returns to discuss the situation with a human.

## 5. NLP Dialog: ALFRED

Interactions of Robbie with humans can follow one of two design options: either train the humans to use robot-friendly commands, or train Robbie to use natural language. While the former is certainly feasible for experts in the robot's language, the latter is desirable for ease of use and maximizing potential sources of information. Successful dialog management is also heavily reliant on metacognition (Anderson & Lee, 2005; McRoy, 1993) as well as learning (Rieger,

1974), in that general learning strategies can also apply to resolving anomalies in conversation. For instance, Robbie may not have understood the task specification, perhaps because of an unusual title (e.g., Smullyan's book "What is the name of this book?") or perhaps because we mumbled. Or, we may have summoned Robbie but fail to notice she has arrived, while she stands waiting for instructions. All of these scenarios will require some sort of dialog-specific reasoning strategies, which allow us to specify expectations and recovery strategies similar to those used in general reasoning. Towards this end, we have been working on a dialog agent named ALFRED (Active Logic For Reason-Enhanced Dialog).

ALFRED is a dialog agent which acts as an interface between a human user and a task-oriented domain (Josyula, 2005). It accepts English sentences as input and parses them into appropriate commands, based on the particular domain and information in its knowledge base (KB). ALFRED is designed to be a general agent and flexible enough to handle a variety of different scenarios. For each domain, ALFRED has a dictionary listing the possible commands and objects, as well as specifying the command syntax for that domain. To implement the NAG-cycle, ALFRED maintains a set of expectations regarding content, time and feedback. That is, it tracks what predicates are expected, when those predicates are expected, and the expected values of parameters in each predicate.

When an expectation is not met, ALFRED interprets it as an indication of an anomaly: noting the problem, assessing the situation, and guiding a response strategy into place. Some examples of responses might be to ignore, try again, adjust the plan or expectation, or ask for help. When the expectation is met, the corresponding violation is removed from the knowledge base; when it is not, ALFRED will attempt another response strategy until the issue is resolved or until it chooses *ignoring the violation* as the response strategy. We are currently conducting experiments that will compare the performance of two quite different implementations of MCL on a variety of different types of anomalies within ALFRED (McNany, Josyula, Cox, Paisner, Perlis, 2013). The two implementations differ in terms of how much of the host KB is shared by MCL. In one setup, generalized MCL is used as a monitor and control mechanism that runs external to the cognitive sphere of the host agent and hence any knowledge sharing between Alfred and generalized MCL is done explicitly. In the other experimental setup, a specialized MCL runs alongside the cognitive reasoner within the same active logic engine (and hence MCL has full access to the KB) to monitor and control the cognitive behaviors of the agent.

An example of metacognition within dialog involves using ALFRED as an interface to direct trains. In this setting, we have implemented the monitoring of the success of initiated responses and evaluation of candidate options before immediately initiating the same response again. For instance, if a user requests "send the Chicago train to New York", ALFRED may choose Metroliner as the candidate, a train that is currently in Chicago. However, if the user replies "No" and repeats the same request, ALFRED evaluates its options, notices that its previous first choice of Metroliner was an unsuccessful response, and instead chooses Northstar, a train that originates in Chicago. In this way ALFRED is able to learn which entity is meant by "the Chicago train" instead of repeatedly choosing the same, incorrect train as a response to the user's request.

## 6. Reinforcement Learning

Reinforcement Learning (RL) allows a robot to learn from an unknown environment. As mentioned in the running example, at first, Robbie knew very little. Therefore, all of her acquired

knowledge about the world and herself are obtained via learning in one form or another. For instance, Robbie has previously learned a floor plan that she now consults to plan a path; Robbie tries the knob and cannot manage to turn it as she has been trained only on door handles; a knob is too hard for her grippers to manage, but if she uses a key, the door will open without having to turn the knob at all. That is, Robbie needs to learn at least to plan a path, to turn a handle, and to use a key to turn a knob. In the remainder of this section we illustrate one particular way that MCL can impact RL methods..

Consider a simple experiment with a standard RL algorithm (Q-learning; see Anderson & Perlis, 2005 for details). An agent is envisioned to maneuver within an 8x8 grid world, in which there is a positive reward in one corner, and negative in the opposite corner. The learner executes 10,000 turns, learning a very effective policy for maximizing reward, as is standard for this sort of learning algorithm. Then we switched the rewards and let the learner continue as before, for an additional 10,000 turns. Not unexpectedly, (see Figure 2) performance degraded. But what is striking is that it recovered far more slowly than it had done in the first 10,000 turns. In effect, it needed to “unlearn” what it had learned before it could then learn the new reward structure.

This of course is not very intelligent. A smarter agent would soon realize that its well-learned strategy no longer worked at all, would stop using it, and would start running the reinforcement over from scratch. When we configured the agent to do that, it learned the new reward structure much faster (see Figure 3 where both old and new methods are superimposed).

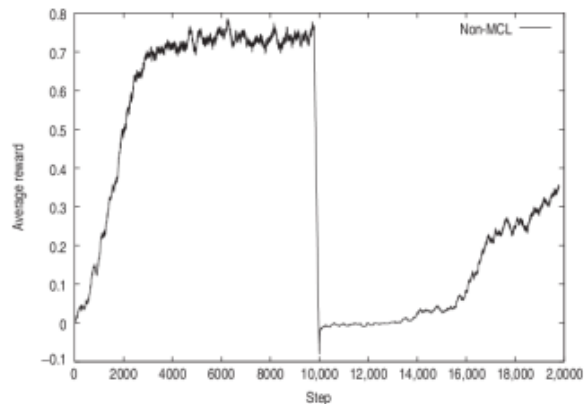


Figure 2. Performance of Q-learner, over 20,000 turns, with initial reward structure of [-10 10] and a post-perturbation reward structure of [10 -10]

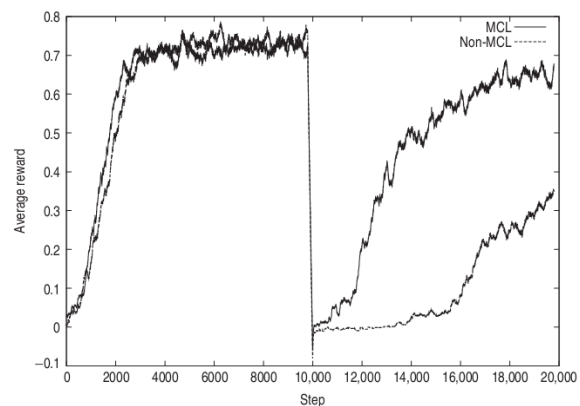


Figure 3. Performance of same Q-learner but with MCL, superimposed on Figure 2

We anticipate that MCL can similarly be used to enhance many kinds of RL and other system components.

## 7. Goal-Driven Autonomy and an Integrated Metacognitive Architecture

*As Robbie leaves our office, it dawns on her that the failure of delivering the book may reoccur the next time she is asked for another book and that even humans may have difficulty finding books they want if they are out of order. Given that she has nothing to do at the moment, she*



*decides to clean up the library and reorganize the book shelves. She remembers being scolded once for borrowing a book without asking first, so she returns to ask our permission. But we are not there, having gone to a noon appointment. Robbie reflects and then decides that rearranging is not the same as borrowing, and that it would be ok to organize the books.*

Autonomy has long been viewed as effectively performing tasks to automatically achieve the goals given to an agent by a human and learning to improve such performance in the future. But a new model of agent autonomy called *goal-driven autonomy* (GDA) asserts that autonomy is also about recognizing novel problems, explaining what caused such problems, and generating one's own goals to solve the problems (Cox, 2007; 2013; Klenk, Molineaux, & Aha, 2013; Maynard, Cox, Paisner, & Perlis, in press). As such this is a variation of the note-assess-guide procedure.

A GDA agent notes when failures occur (Robbie sought to achieve her goal but did not succeed), assesses the failure (Robbie failed because the book was not shelved correctly), and then guides a response to the failure (Robbie generated a goal to correctly shelve the books). Here the note phase is similar: an observation does not match the expectation and hence a discrepancy (i.e., anomaly) exists. However the assess phase involves determining a causal explanation for the failure or discrepancy. The response is to generate a new goal to solve the problem. The generation of such goals can be found by determining a salient antecedent of the explanation and negating it (Cox, 2007; 2013). Here the robot generates the goal of not having the books being shelved incorrectly.

We have been working on implementing a larger cognitive architecture that integrates much of this work. The *Metacognitive, Integrated, Dual-Cycle Architecture (MIDCA)* (Cox, Maynard, Paisner, Perlis, & Oates, 2013; 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 formation, 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 world 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. 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.

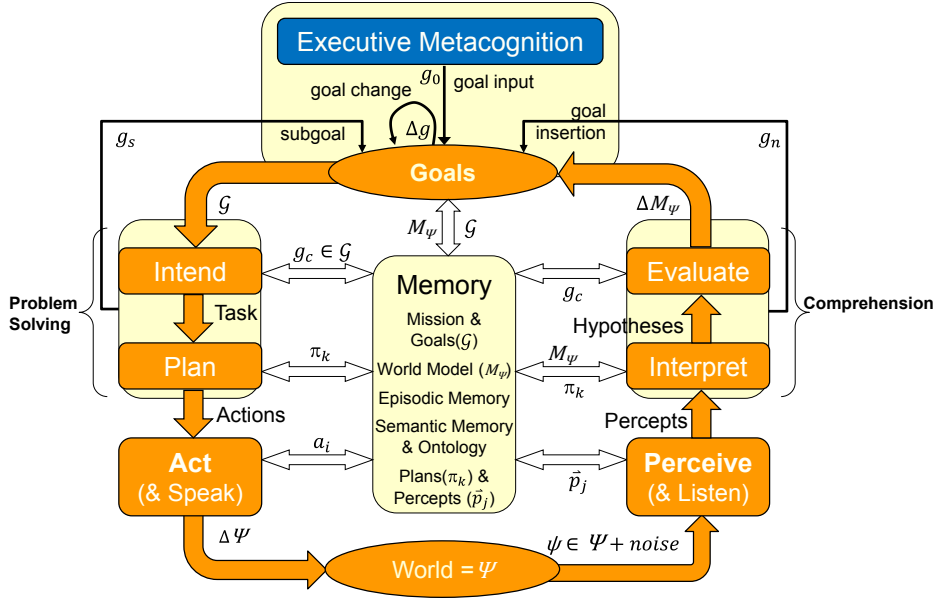


Figure 4. Object-level detail with meta-level goal management

To illustrate these distinctions, consider the object level as shown in Figure 4. 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 ( $\Delta g$ ). 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 ( $\pi_k$ , e.g., a hierarchical-goal-net plan Shivashankar, Kuter, Nau, & Alford, 2012; Shivashankar, Alford, Kuter, & Nau, 2013) that instantiates that task given the current model of the world ( $M_\psi$ ) and its background knowledge (e.g., semantic memory and ontologies). The plan is executed to change the actual world ( $\Psi$ ) through the effects of the planned actions ( $a_i$ ). The goal and plan are stored in memory and constitute the agent’s 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 the world with respect to the goal.

Comprehension starts with perception of the world in the attentional field. Interpretation takes as input the resulting percepts ( $\vec{p}_j$ ) and the expectations in memory ( $\pi_k$  and  $g_c$ ) to determine whether the agent is making sufficient progress. An MCL note-assess-guide procedure 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 ( $g_n$ ). In the absence of an anomaly, the agent incorporates the changes inferred from the percepts into the world model ( $\Delta M_\psi$ ) 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.

## 8. Memory

Memory is a crucial function for a perpetual cognitive agent. Conceptual information in a *semantic memory* is functionally important for interpreting perceptions and for reasoning about the world. But equally important, an *episodic memory* stores a personal history of the agent and its interactions with the world and other agents in that world (e.g., see Laird, Nuxoll, & Derbinsky 2012). If the agent is to reason about its capacity to perform actions in the present, it is important that it knows what worked and did not work in the past. It is not always possible to infer results, especially when it comes to the behavior of humans; so for instance, remembering the likes and dislikes of others (e.g., remembering that asking permission is important to someone) is a useful function. Furthermore, rather than having to solve problems from scratch each time, an agent should remember how it solved similar problems in the past and simply reuse the past solution or adapt an old solution to fit new circumstances. Such a case-based reasoning approach (de Mántaras, et al., 2006; Kolodner, 1993) has been shown to reduce effort and make for efficient problem solving (e.g., Cox, Munoz-Avila, & Bergmann, 2006; Veloso, 1994).

Memory should organize conceptual and procedural information in a manner that makes it effective. A good memory retains useful information and makes it available at the right time in the right form (Schank, 1982). Memory in cognitive agents can be partitioned into separate functions and controlled by an inference cycle mechanism; e.g., see (Elgot-Drapkin, Miller, & Perlis, 1991) in which preliminary experiments illustrated a critical impact of the size of working memory/STM. The benefit of a good memory architecture is that knowledge need not be searched by an arbitrary brute-force approach; rather an agent can depend upon a retrieval match between a contextual cue and the index used to store a memory. The cost is in terms of what has been called the indexing problem (Kolodner, 1993; Schank, 1982; Schank & Osgood, 1990). The problem is to choose effective cues, or features in an input, to be used as probes for retrieving from memory the knowledge structures necessary to process an input.

The converse problem is the problem of forgetting (Cox, 1994). If the cues are not chosen with care during retrieval time, or if the indexes are not chosen well during encoding, the reasoner may not recall a memory structure when it is needed. The forgetting problem is to reorganize memory and the indexes by which memory is accessed. Because reasoning failures may occur due to faulty memory organization, as well as because of faulty reasoning components or faulty knowledge, the selection or retrieval of knowledge plays an important role in determining the cause of failure.

Reasoning failures related to information retrieval have been addressed using metamemory (Caro, Jimenez, & Paternina, 2012; Leake, 1995). In artificial intelligence, metamemory refers to the processes and techniques a system uses to monitor and control its own memory, which has strong parallels in cognitive psychology research (Nelson, Narens, & Dunlosky, 2004; Metcalfe & Dunlosky, 2008). Indeed, to realize one's own memory limitations – no memory architecture will be perfect in retrieving exactly the right information at the optimal moment – is an important piece of self-knowledge that can guide an agent's behavior (e.g., setting reminders for itself). Keeping all information – even just all episodic information – in working memory where the agent's reasoning processes can run rampant on it (the swamping or omniscience problem) will only clog the agent's ability to act in a timely way. So metamemory can have a role in helping an agent mark items that are important to retain for a time (Robbie marks the book-title information that way) and allow others to be gracefully “forgotten.”

## 9. Related Work

Since McCarthy originally described the concept of a computer advice taker (McCarthy, 1959), many research projects have embraced the goal of implementing persistent agents that co-exist and interact with humans over extended time periods. The original work at SRI on Shakey the robot (Nilsson, 1984) combined sub-systems that reasoned using logic and that acted through sensors and effectors on the platform. It represented the first effort to build a human-like intelligent physical and mental system (i.e., an instantiation of sorts of Nilsson's, 1983, *computer individual*), although it was extremely brittle given any unforeseen circumstances.

Our idea of a perpetual cognitive real-world agent relates to variations on 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; Scassellati, 2001), and developmental cognitive robots that learn over time (e.g., Weng et al., 2001). Researchers have approached the research in various ways resulting in theories of human-level intelligence (Cassimatis, 2012; Cassimatis & Winston, 2004) and artificial general intelligence (Wang & Goertzel, 2012; Crowder & Friess, 2010).

A number of more recent research projects exist within the artificial intelligence and cognitive science communities that integrate multiple high-level cognitive functions and perform complex tasks in dynamic environments, some with actual physical platforms or robots. Well known examples include ACT-R (Anderson & Lebiere, 1998), CogAff (Sloman, 2003, 2011), Companion Cognitive Systems (Forbus, Klenk, & Hinrichs, 2009), EM-One (Singh, 2005), DIARC (Krause, Schermerhorn, & Scheutz, 2012), EPILOG (Morbini & Schubert, 2011), Icarus (Langley & Choi, 2006), SALS (Morgan, 2009), SNePS (Shapiro, 2000), Soar (Laird, 2012), and SS-RICS (Kelley, 2003).

## 10. Current and Future Plans and Conclusion

We are currently working on a variety of additional aspects of our long-term goal of human-level autonomous systems. One large portion of the work involves combining Alfred/Active Logic with MCL; as a particular example, we are investigating metacognitive means to allow a conversational agent to deal with unanticipated pauses in a conversation (McNany, E., Josyula, D., Cox, M. T., Paisner, M., Perlis, D. (2013)). With regard to reinforcement learning, we are exploring the use of the natural-actor critic algorithm in conjunction with the growing neural gas (GNG) algorithm to help a robot learn the effects of its actions, like a baby thrashing about until it learns that motor impulses and visual and tactile inputs correlate in highly regular ways. And with regard to MIDCA itself, we are using a symbolic version of the A-distance algorithm along with GNG to help a system identify (note) anomalies (Paisner, Perlis, & Cox, in press).

We believe that a frontal assault on the challenge of human-level AI is timely, that many of the needed tools are currently available, and that many of the very real remaining gaps can be filled along the lines we have sketched here. One particular thrust that we envision is that of a competitive robot treasure-hunt with, say, PR2 robots (that have a considerable degree of fine manipulative capacity as well as ease of programming). The treasure-hunt domain nicely combines natural-language, perception, real-time planning, goal-creation, action, and indeed most of AI.

## Acknowledgements

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, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, N. J: LEA, 1998.
- Anderson, M. L., Josyula, D., Okamoto, Y., & Perlis, D. (2002). Time-situated agency: Active logic and intention formation. In *Proceedings of the 25th German Conference on Artificial Intelligence*.
- 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., & Lee, B. (2005). Metalanguage for dialog management. I. In *Proceedings of the 16th Annual Winter Conference on Discourse, Text and Cognition*.
- 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.
- 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.
- Caro, M., Jimenez, J., & Paternina, A. (2012). Architectural modeling of metamemory judgment in case-based reasoning systems. *2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)* (pp. 1–8).
- Cassimatis, N. L. (2012). Human-level artificial intelligence must be an extraordinary science *Advances in Cognitive Systems I*, 37- 45.
- 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.
- Cox, M. T. (1994). Machines that forget: Learning from retrieval failure of mis-indexed explanations. In A. Ram & K. Eiselt (Eds.), *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* (pp. 225-230). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI Magazine* 28(1), 32-45.
- Cox, M. T. (2013). *Goal-driven autonomy and question-based problem recognition*. Submitted.
- Cox, M. T., Maynard, 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.
- Cox, M. T., Munoz-Avila, H., & Bergmann, R. (2006). Case-based planning. *Knowledge Engineering Review* 20(3), 283-287.

- 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.
- Crowder, J. A., & Friess, S. (2010). Metacognition and metamemory concepts for AI systems. In *International Conference on Artificial Intelligence, ICAI 2010*. Las Vegas.
- 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.
- De Mántaras, R. L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., Aamodt, A., & Watson, I. (2006). Retrieval, reuse and retention in case-based reasoning. *Knowledge Engineering Review*. 20(3), 215-240.
- Derbinsky, N., & Laird, J. E. (2010). Extending soar with dissociated symbolic memories. In M. Lim & W. Ho (Eds.), *Proceedings of the Remembering Who We Are: Human Memory for Artificial Agents Symposium*, At the AISB 2010 convention, De Montfort Univ., Leicester, UK.
- Elgot-Drapkin, J., Miller, M., & Perlis, D. (1991). Memory, reason, and time: The step-logic approach. In Cummins and Pollock (Eds.), *Philosophy and AI: Essays at the interface*. Cambridge, MA: 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.
- Josyula, D. (2005). *A unified theory of acting and agency for a universal interfacing agent*, Ph.D. dissertation, University of Maryland, College Park.
- Josyula, D and M'Bale, K. (2013). Bounded Metacognition. In *Proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications* (pp.147-152). Red Hook, NY: Curran Associates.
- Kelley, T. (2003). Symbolic and sub-symbolic representations in computational models of human cognition: What can be learned from biology? *Theory and Psychology*, 13(6), Dec. 2003.
- Klenk, M., Molineaux, M., & Aha, D. (2013) Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence* 29(2), 187–206.
- Kolodner, J. L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.
- 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.
- Laird, J. E. (2012). *The Soar Cognitive architecture*. Cambridge, MA: MIT Press.
- Laird, J. E., Nuxoll, A., & Derbinsky, N. (2012). Episodic memory. In J. Laird, *The Soar cognitive architecture* (pp. 225-246). Cambridge, MA: MIT Press.
- Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. *The Twenty-First National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.
- Leake, D. B. (1995). Representing Self-knowledge for Introspection about Memory Search A Planful Framework for Internal Reasoning. In M. T. Cox & M. Freed (Eds.), *AAAI Spring Symposium on Representing Mental States and Mechanisms*. Stanford, CA.

- Maynard, 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.
- 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.
- McNany, E., Josyula, D., Cox, M. T., Paisner, M., Perlis, D. (2013). Metacognitive guidance in a dialog agent. In *Proceedings of the Fifth International Conference on Advanced Cognitive Technologies and Applications* (pp.137-140). Red Hook, NY: Curran Associates.
- McRoy, S. (1993). *Abductive interpretation and reinterpretation of natural language utterances*. Ph.D. dissertation, University of Toronto, 1993.
- Metcalfe, J., & Dunlosky, J. (2008). Metamemory. In H. L. Roediger III (Ed.), *Cognitive Psychology of Memory. Vol. 2 of Learning and Memory: A comprehensive reference*, 4 vols.
- 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.
- Morgan, B. (2009). Funk2: A distributed processing language for reflective tracing of a large critic-selector cognitive architecture. In *Proceedings of the Metacognition Workshop at the Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*. San Francisco, CA.
- Nelson, T., Narens, L., & Dunlosky, J. (2004). A revised methodology for research on metamemory: Pre-judgment recall and monitoring (PRAM). *Psychological methods*, 9(1), 53–69.
- Nilsson, N. (1983). Artificial intelligence prepares for 2001. *AI Magazine* 4(4), 7-14.
- Nilsson, N. (1984). *Shakey the robot*. Technical Note 323, Artificial Intelligence Center, Menlo Park, CA: SRI.
- 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.
- Purang, K. (2001) Alma/Carne: Implementation of a Time-Situated Meta-Reasoner. In *Proceedings of the International Conference on Tools with Artificial Intelligence*, 103-110.
- Rieger, C. (1974). *Conceptual memory: A theory and computer program for processing the meaning content of natural-language utterances*. Ph.D. dissertation, Stanford University, 1974.
- Scassellati, B. M. (2001). *Foundations for a theory of mind for a humanoid robot*. Doctoral dissertation, Massachusetts Institute of Technology, Cambridge.
- Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge, MA: Cambridge University Press.
- Schank, R. C., & Osgood, R. (1990). *A content theory of memory indexing*. Technical Report 2. Institute for the Learning Sciences, Northwestern University, Evanston, IL.
- 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*. Los Alamitos, CA: IEEE.
- Shapiro, S. (2000). SNePS: A logic for natural language understanding and commonsense reasoning. In L. Iwanska & S. Shapiro (Eds.), *Natural language processing and knowledge representation: Language for knowledge and knowledge for language* (pp. 175–195). Menlo Park, CA: AAAI Press/The MIT Press.
- Shivashankar, V., Alford, R., Kuter, U., & Nau, D. (2013). The GoDeL planning system: A more perfect union of domain-independent and hierarchical planning. In *Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI-13)*.
- Shivashankar, V., Kuter, U., Nau, D., & Alford, R. (2012). A hierarchical goal-based formalism and algorithm for single-agent planning. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Schmill, M., Anderson, M., Fults, S., Josyula, D., Oates, T., Perlis, D., Haidarian, S., & Wilson, S. The metacognitive loop and reasoning about mistakes. In M. T. Cox & A. Raja (eds), *Thinking about thinking*, MIT Press, 2011.
- Singh, P. (2005). *EM-ONE: An architecture for reflective commonsense thinking*. Ph.D. dissertation. Department of Electrical Engineering and Computer Science. MIT. Boston, MA.
- Sloman, A. (2003). *The Cognition and Affect project: Architectures, architecture-schemas, and the new science of mind* (Technical Report). Birmingham, UK, School of Computer Science, University of Birmingham.
- 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.
- Veloso, M. (1994). *Planning and learning by analogical reasoning*. Berlin: Springer-Verlag.
- 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.