

Perpetual Self-Aware Cognitive Agents

Michael T. Cox

■ To construct a perpetual self-aware cognitive agent that can continuously operate with independence, an introspective machine must be produced. To assemble such an agent, it is necessary to perform a full integration of cognition (planning, understanding, and learning) and metacognition (control and monitoring of cognition) with intelligent behaviors. The failure to do this completely is why similar, more limited efforts have not succeeded in the past. I outline some key computational requirements of metacognition by describing a multi-strategy learning system called Meta-AQUA and then discuss an integration of Meta-AQUA with a nonlinear state-space planning agent. I show how the resultant system, INTRO, can independently generate its own goals, and I relate this work to the general issue of self-awareness by machine.

Although by definition all AI systems can perform intelligent activities, virtually none can understand why they do what they do, nor how. Many research projects have sought to develop machines with some metacognitive capacity, yet until recently no effort has attempted to implement a complete, fully integrated, metalevel architecture. Many in the AI community argue that base-level cognition in isolation is insufficient and that a fully situated agent is necessary. Similarly I claim that a metalevel cognitive theory must be comprehensive if it is to be successful. Previous attempts, including the research of Cox and Ram (1999a, Cox 1996b), Fox and Leake (1995, Fox 1996), and Murdock and Goel (2001, Murdock 2001), to build introspective agents have been insufficient because of their limited extent and scope. This article examines a broader approach to the design of an agent

that understands itself as well as the world around it in a meaningful way.¹

Figure 1 shows a decomposition of the interrelationships among problem solving, comprehension, and learning. These reasoning processes share a number of intersecting characteristics. As indicated by the intersection labeled *A* on the lower left, learning can be thought of as a deliberate planning task with its own set of *learning goals* (Cox and Ram 1995). Instead of a goal to alter the world such as having block *X* on top of block *Y*, a learning goal might be to obtain a semantic segregation between concepts *X* and *Y* by altering the background knowledge through a learning plan. As indicated by the *B* intersection of figure 1, the learning process and the story understanding task within natural language processing share many traits (Cox and Ram 1999b). In both explanation is central. To understand a story completely, it is necessary to explain unusual or surprising events and to link them into a causal interpretation that provides the motivations and intent supporting the actions of characters in the story. To learn effectively an agent must be able to explain performance failures by generating the causal factors that led to error so that similar problems can be avoided in the future. Here I discuss in some detail issues related to the letter *D* intersection.²

The area labeled *D* represents the intersection of planning and comprehension, normally studied separately. Planning is more than generating a sequence of actions that if executed will transform some initial state into a given goal state. Instead planning is embedded in a larger plan-management process that must interleave planning, execution, and plan understanding (Chien et al. 1996, Pollack and Horty 1999). Furthermore while many AI systems accept goals as input or have inherent background goals that drive system behavior,

few if any systems strategically derive their own explicit goals given an understanding (that is, comprehension) of the environment.

A major objective of this article is to show how a preliminary system called INTRO (*initial introspective cognitive-agent*) can systematically determine its own goals by interpreting and explaining unusual events or states of the world. The resulting goals seek to change the world in order to lower the dissonance between what it expects and the way the world is. The mechanism that INTRO uses for explanation is the same as the mechanism its metareasoning component uses when explaining a reasoning failure. The resulting goals in the latter case seek to change its knowledge in order to reduce the chance of repeating the reasoning failure (that is, a learning goal is to change the dissonance between what it knows and what it should know).

I begin by describing the Meta-AQUA comprehension component that implements a theory of introspective multistrategy learning. The general idea is that an agent needs to represent a *naive theory of mind* to reason about itself. It does not need a complete and consistent axiomization of all mental activities and computations; rather it need only represent basic, abstract patterns within its mental life like “I was late for my appointment, because I forgot where I parked the car.” Next I present the PRODIGY planning component that creates action sequences for INTRO to perform in the Wumpus World simulated environment. I then examine the current implementation of the INTRO cognitive agent and follow with a section that discusses the difficult choice an agent has between doing something or learning something in the face of anomalous input. I conclude by considering what it might mean for an artificial agent to be self-aware.

Self-Understanding and Meta-AQUA

Meta-AQUA³ (Cox 1996b; Cox and Ram 1999a; Lee and Cox 2002) is an introspective multistrategy learning system that improves its story-understanding performance through a metacognitive analysis of its own reasoning failures. The system’s natural language performance task is to “understand” stories by building causal explanatory graphs that link the individual subgraph representations of the events into a coherent whole where an example measure of coherency is minimizing the number of connected components. The performance subsystem uses a multistrategy approach to comprehension. Thus, the top-level

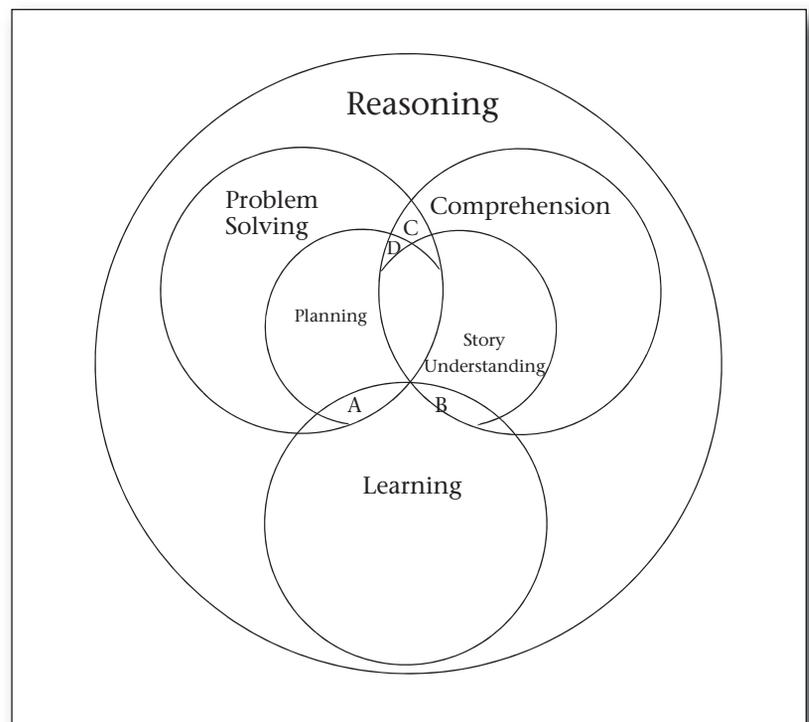


Figure 1. Hierarchical Decomposition of Reasoning.

goal is to choose a comprehension method (for example, script processing, case-based reasoning, or explanation generation) by which it can understand an input. When an anomalous or otherwise interesting input is detected, the system builds an explanation of the event, incorporating it into the preexisting model of the story. Meta-AQUA uses case-based knowledge representations implemented as frames (Cox 1997) tied together by *explanation-patterns* (Schank 1986; Schank, Kass, and Riesbeck 1994; Ram 1993) that represent general causal structures.

Meta-AQUA uses an interest-driven, variable depth, interpretation process that controls the amount of computational resources applied to the comprehension task. Consider the example in figure 2. Sentences S1 through S3 generate no interest, because they represent normal actions that an agent does on a regular basis. But Meta-AQUA classifies S4 to be a violent action and, thus according to its interest criterion (discussed in more detail later), interesting. It tries to explain the action by hypothesizing that the agent shot the Wumpus because she wanted to practice sport shooting. An abstract explanation pattern, or XP, retrieved from memory instantiates this explanation, and the system incorporates it into the current model of the actions in the story. In sentence S6, however, the story specifies an alternate

S1: The Agent left home.
 S2: She traveled down the lower path through the forest.
 S3: At the end she swung left.
 S4: Drawing her arrow she shot the Wumpus.
 S5: It screamed.
 S6: She shot the Wumpus, because it threatened her.

Figure 2. The Wumpus Story.

	Description	Correspondence to Figure 3
Failure Symptoms	Contradiction between expected explanation and actual explanation.	$(E = \text{Wumpus-as-target-XP})^1 \neq (A_2 = \text{Wumpus-as-threat-XP})$
Faults	Novel situation. Erroneous association.	$\text{out}_{BK}(M')^*$ index: $l = \text{physical-obj.}$
Learning Goals	Segregate competing explanations. Acquire new explanation.	goal 1: G1 goal 2: G2
Learning Plan	Generalize threat explanation. Store and index new explanation. Mutually reindex two explanations.	specific threat explanation: A_2 general threat explanation: E' memory items: M and M'
Plan Execution Results	New general case of shoot explanation acquired. Index new explanation. Reindex old explanation.	$\text{generalize}(A_2) \Rightarrow E'$ $\text{store}(E') \Rightarrow M'$ $\text{index}(M') \Rightarrow l' = \text{inanimate-obj}$ $\text{index}(M) \Rightarrow l = \text{animate-obj}$

Table 1. Learning from Explanation Failure.

* Out of the set of beliefs with respect to the background knowledge (Cox 1996b, Cox and Ram 1999a).

explanation (that is, the shoot action is in response to a threat). This input triggers an expectation failure, because the system had expected one explanation to be the case, but another proved true instead.

When Meta-AQUA detects an explanation failure such as this one, the performance module passes a trace of the prior reasoning to the learning subsystem. Three sequences of events then need to occur for effective learning to take place. First, it must explain the failure by mapping the failure symptom to the causal fault. Second, it must decide what to learn by generating a set of learning goals. Third, it must construct a learning strategy by planning to achieve the goal set.

To explain why the failure occurred, Meta-AQUA assigns blame by applying an introspective explanation to the reasoning trace. A *meta-explanation pattern* (Meta-XP) is developed by

retrieving an XP using the failure symptom as a probe into memory. Meta-AQUA instantiates the retrieved meta-explanation and binds it to the trace of reasoning that preceded the failure. The resulting structure (see figure 3) is then checked for applicability. If the Meta-XP does not apply correctly, then another probe is attempted. An accepted Meta-XP either provides a set of learning goals (determines what to learn) that are designed to modify the system's background knowledge or generates additional questions to be posed about the failure. Once a set of learning goals is posted, they are passed to a nonlinear planner for building a learning plan (strategy construction).

Figure 3 represents the following pattern. Meta-AQUA posed an initial question. In this case the question was "Why did the agent shoot the Wumpus?" A memory probe returned an inappropriate explanation that con-

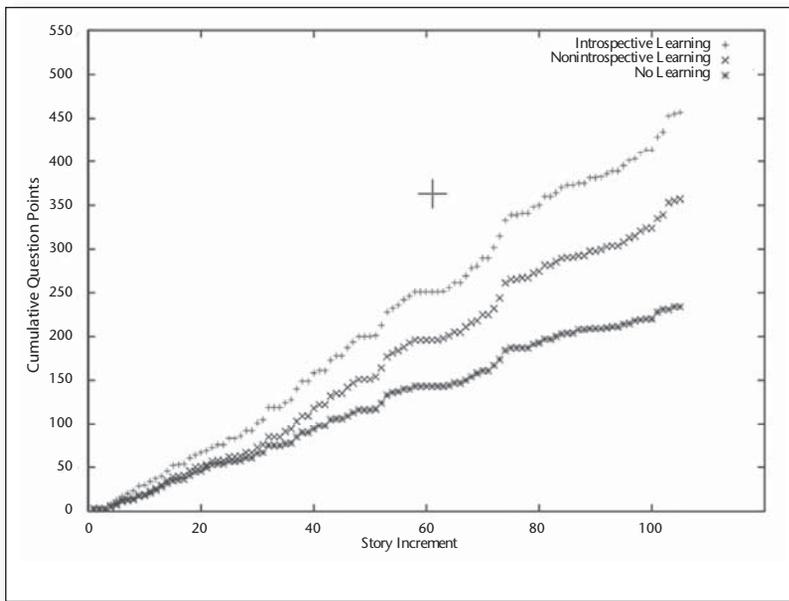


Figure 4. Cumulative Explanation Performance as a Function of Metacognitive Reasoning.

tem scored points for each correct answer. Further details of the experiment can be found in Cox and Ram (1999a). The data in figure 4 show that the approach sketched here outperforms an alternative that does not use the learning goals mechanism, and the alternate outperforms no learning at all. However, Cox and Ram (1999a) report specific cases in these experiments where learning algorithms negatively interact and thus nonintrospective learning actually performs worse than the no learning condition.

Given that the XP application algorithm is involved in both, Meta-AQUA understands itself as it understands stories. That is, the AQUA system (Ram 1991, 1993), a microversion of which exists as the performance component within Meta-AQUA, explains explicitly represented stories using the XP application. Meta-AQUA explicitly represents traces of the explanation process and uses the algorithm to explain explanation failure and, thus, to understand itself through meta-explanation.

Awareness and PRODIGY

The PRODIGY⁵ planning and learning architecture (Carbonell, Knoblock, and Minton 1991; Veloso et al. 1995) implements a general problem-solving mechanism that builds a sequence of actions given a domain description, initial state, and goal conjunct. At its core is a nonlinear state-space planner called Prodi-

gy4.0. It follows a means-ends analysis backward-chaining search procedure that reasons about both multiple goals and multiple alternative operators from its domain theory. PRODIGY uses a STRIPS-like operator representation language whose Backus-Naur form (BNF) is provided in Carbonell et al. (1992). PRODIGY is actually a set of programs, each of which explores a different facet of the planning and learning processes.

The core generative planner is currently Prodigy4.0. It uses the following four-step decision process to establish a sequence of actions that transforms an initial state of the world into the given goal state: (1) It selects a goal to solve from a list of candidate goals; (2) it selects an operator from a list of candidate operators that can achieve the selected goal; (3) it selects object bindings for open operator variables from a set of candidate bindings; and (4) if instantiated operators exist having no open preconditions and pending goals also exist, it chooses either to apply the instantiated operator (forward chain) or to continue subgoaling (backward chain).

A set of planning operators compose the representation of action in the domain. Table 2 shows an operator that enables PRODIGY to navigate forward in a grid environment where all movement is constrained along the directions north, south, east, and west. The operator contains four variables: an agent; a start location and a destination location; and a facing orientation for the agent. The operator preconditions specify that the agent is at the start location and that the start and destination are traversable. The facing and is-loc predicates constitute conditions that constrain search given a set of static domain axioms such as (is-loc Loc12 north Loc11) for all adjacent locations in the environment. An add list and a delete list represent the effects of the operator by changing the agent's location from the start to the destination. PRODIGY navigates from one location to another by recursively binding the goal location with the destination variable and subgoaling until the destination can be bound with an adjacent cell to the current location of the agent.

Prodigy/Analogy (Veloso 1994) implements a case-based planning mechanism (Cox, Muñoz-Avila, and Bergmann 2006) on top of the Prodigy4.0 generative planner. The main principle of case-based or analogical reasoning is to reuse past experience instead of performing a cognitive task from scratch each and every time a class of problems arises. This principle applies to both comprehension tasks such as was the case with explanation in Meta-

```

(OPERATOR FORWARD
 (params <agent1> <location1> <location2>)
 (preconds
  ((<agent1> AGENT)
   (<location1> LOCATION)
   (<location2> (and LOCATION
                    (diff <location1> <location2>))))
   (<orient1> ORIENTATION))
 (and
  (traversable <location1>)
  (traversable <location2>)
  (at-agent <agent1> <location1>)
  (facing <orient1>)
  (is-loc <location2> <orient1> <location1>)))
 (effects
  ()
  ((del (at-agent <agent1> <location1>))
   (add (at-agent <agent1> <location2>))))))

```

Table 2. PRODIGY Operator for Forward Movement in a Grid Environment.

AQUA and to problem-solving tasks such as planning. The task in case-based planning is to find an old solution to a past problem that is similar to the current problem and to *adapt* the old plan so that it transforms the current initial state into the current goal state (Muñoz-Avila and Cox 2006). In Prodigy/Analogy the case-based approach takes an especially appropriate direction relative to self-aware systems.

Carbonell (1986) argues that two types of analogical mappings exist when using past experience in planning. Transformation analogy directly changes an old plan to fit the new situation so as to achieve the current goals. Alternatively derivational analogy uses a representation of the prior reasoning that produced the old plan to reason likewise in the new context. Prodigy/Analogy implements derivational analogy by annotating plan solutions with the plan justifications at case storage time and, given a new goal conjunct, replaying the derivation to reconstruct the old line of reasoning at planning time (Velo and Carbonell 1994). Figure 5 illustrates the replay process in the Wumpus World domain from the PRODIGY User Interface 2.0 (Cox and Velo 1997). Velo (1994) has shown significant improvement in performance over generative planning using such plan rationale.

The justification structures saved by Prodigy/Analogy begin to provide a principled mechanism that supports an agent's awareness of what it is doing and why. In particular the representation provides the goal-subgoal structure relevant to each action in the final plan. But beyond recording goal linkage to action selection, Prodigy/Analogy records the decision basis and alternatives for each of the four decision points in its planning process. Like Meta-AQUA Prodigy/Analogy can reason about its own reasoning to improve its performance. One difference is that Prodigy/Analogy capitalizes upon success to learn, whereas Meta-AQUA depends upon failure for determining those portions of its knowledge that require improvement. Yet both systems require an outside source to provide the goals that drive the performance task. In both cases the programs halt once the achievement or comprehension goal is solved.

INTRO

This section describes a preliminary implementation of an *initial introspective* cognitive agent called INTRO⁶ that is designed to exist continually and independently in a given environment. INTRO can generate explicit declara-

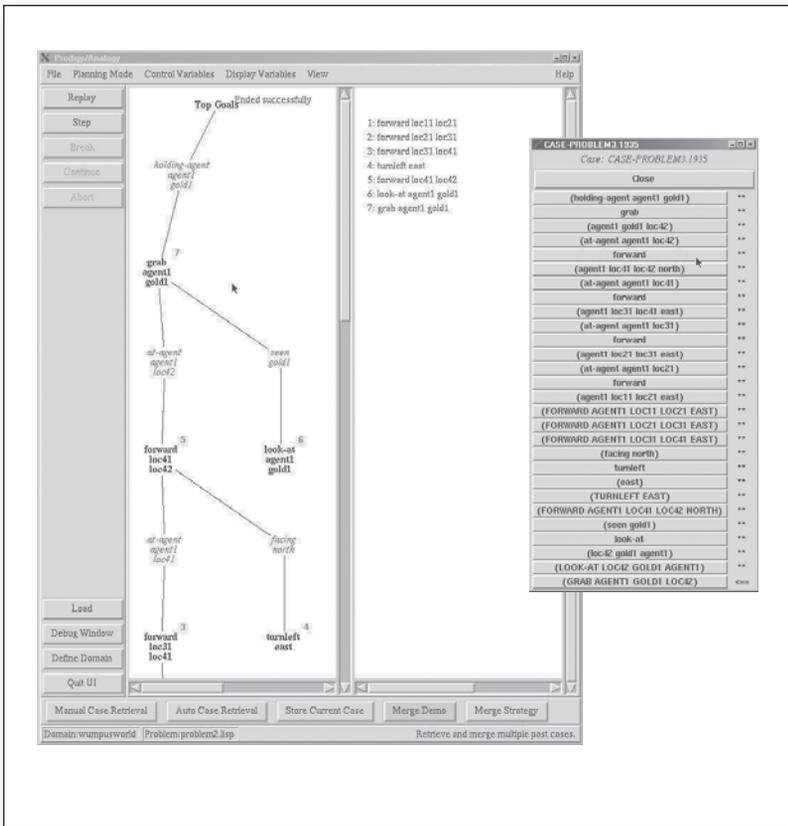


Figure 5. Prodigy/Analogy Replays a Past Derivational Trace of Planning on a New Problem.

tive goals that provide intention and a focus for activities (see Ram and Leake [1995] for a functional motivation for the role of explicit goals). This work is very much in the spirit of some recent research on persistent cognitive assistants (for example, Haigh, Kiff, and Ho [2006]; Myers and Yorke-Smith [2005]), but it examines issues outside the mixed-initiative relationship with a human user.

The agent itself has four major components (see figure 6). INTRO has primitive perceptual and effector subsystems and two cognitive components. The cognitive planning and understanding subsystems consist of the Prodigy/Agent and Meta-AQUA systems respectively. The base cognitive cycle is to observe the world, form a goal to change the world, create a plan to achieve the goal, and finally act in accordance with the plan, observing the results in turn.

The Wumpus World

INTRO operates in a very simple environment described by Russell and Norvig (2003). The Wumpus World⁷ contains an agent whose goal is to find a pot of gold while avoiding pits and the Wumpus creature. Unlike the Wumpus, the

INTRO agent can perform actions to change the environment including turn, move ahead, pick up, and shoot an arrow. Unlike a classical planning domain, the environment is not fully observable, but rather the agent perceives it through percepts. The percepts consist of the 5-tuple [stench, breeze, glitter, bump, scream] that represents whether the Wumpus is nearby, a pit is nearby, gold is colocated with the agent, an obstacle is encountered, and the Wumpus is making a sound.

For the purposes of this example, the environment has been limited to a four by two cell world (see figure 7). The agent, depicted as the character 1, always starts in cell [1,1] facing east at the southwesternmost end of the grid. The Wumpus (W), pits (O), and the gold (\$) can be placed in any of the remaining cells. In a major change to the interpretation of the game, the Wumpus is rather benign and will not injure the agent. It screams when it is hungry and an agent is adjacent, as well as when it is shot with an arrow. The agent can choose either to feed the Wumpus or to shoot it with the arrow. Either will prevent continuous screaming.

The original Wumpus World agent maps a given input tuple to an output action choice by following an interpretation program and a memory of previous percepts and actions. The agent control code was modified to accept instead an action taken from plans output by the Prodigy/Agent component of INTRO. The simulator then simply presents a visualization of the events as the actions execute. As the implementation currently stands, the output of the simulator is not used. Ideally (as shown in the dashed arrow of figure 6) the 5-tuple percept output should be input into the perceptual component. The one exception is that the simulator was changed so that when the agent enters the same grid cell as the Wumpus, the sound state of the Wumpus will change to scream. Code was added so that this action is reported to Meta-AQUA as it occurs.

The Perceptual Subsystem

The perception subsystem [sic] does not model a realistic perceptual filtering of the world and its events. Instead the module in its present form acts as a translator between the representation of Prodigy/Agent and Meta-AQUA. It serves more as a means for input than it does for perception.

The main problem of translation is one of mapping a flat STRIPS operator to an arbitrarily deep, hierarchical, slot-filler event. For example the action FORWARD(agent1,loc11,loc12) must translate into an appropriate hierarchical frame representation. Problems exist

when the parameters do not match in terms of both names and content, when they differ in relative order, or when extraneous parameters exist. For example the Meta-AQUA frame definition in table 3 does not directly correspond to the PRODIGY operator in table 2. To resolve such problems, a mapping function exists for translation.

The Meta-AQUA Comprehension Component

Understanding the actions of a story and the reasons why characters perform such actions is very similar to comprehending the actions and motivations of agents in any environment. So within the INTRO cognitive agent, Meta-AQUA performs this task to understand the results of its own actions using the approach described in the self-understanding and Meta-AQUA section. In both cases Meta-AQUA inputs the events in a conceptual representation and builds an internal model to reflect the causal connections between them. In both cases an anomaly or otherwise interesting event causes Meta-AQUA to generate an explanation of the event. However, instead of using the explanation to modify its knowledge, INTRO uses the explanation to generate a goal to modify the environment. Stated differently, its central role is that of *problem recognition*. That is, how can a persistent agent recognize when a novel problem exists in the world and form a new goal to resolve it?

As is shown in the example output of figure 8, Meta-AQUA processes three forward movements that are not interesting in any significant way. These actions (displayed in the lower right Prodigy/Agent plan window) are therefore skimmed and simply inserted into the model of the Wumpus World actions. However when the system encounters the scream action, it processes it differently, because sex, violence, and loud noises are inherently interesting (Schank 1979).⁸ Notice that Meta-AQUA presents two large shaded windows that display internal representations of the cognitive processing. In the “Goal Monitor” window (far left of figure), the system shows a goal (that is, ID.3206) to identify interesting events in the input. The scream input causes Meta-AQUA to spawn a new goal (that is, GENERATE.3264) to generate an explanation for the scream. This goal is a knowledge goal or question whose answer explains why the Wumpus performed the action (Ram 1991).⁹ The “Memory Monitor” window (upper right) displays a representation of the memory retrieval and storage activities along with the indexes and mental objects that occupy memory.

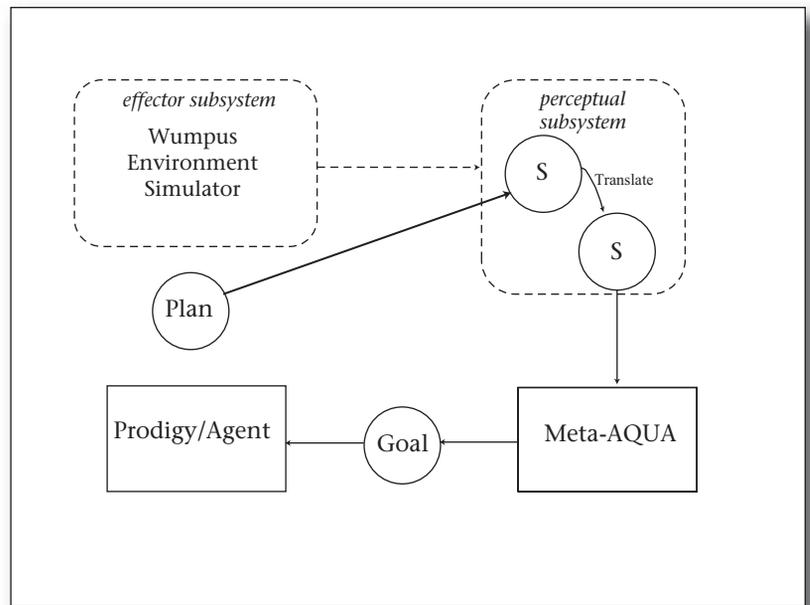


Figure 6. INTRO Architecture.

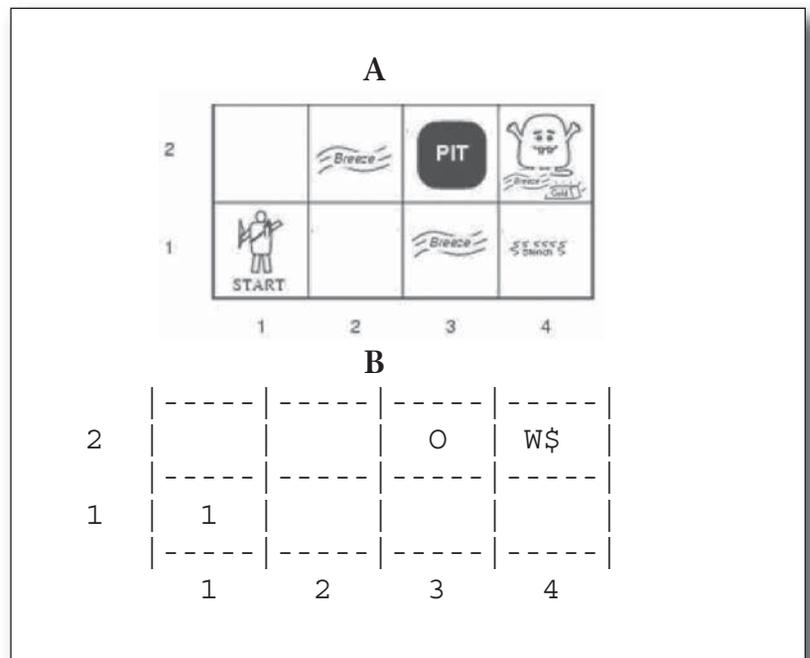


Figure 7. Initial Wumpus World State.

A. Pictorial representation. B. ASCII representation.

```

(define-frame FORWARD
  (isa (value (ptrans)))
  (actor (value (agent)))
  (object (value =actor))
  (from (value (at-location
                (domain (value =actor))
                (co-domain (value (location))))))
  (to (value (at-location
              (domain (value =actor))
              (co-domain (value (location)))))))

```

Table 3. Frame Definition for Forward Movement.

As a result of this activity, the Meta-AQUA component attempts to explain the scream. The background memory contains a conceptual network, a case/script library, a set of causal explanation patterns (XPs) and meta-explanation patterns (Meta-XPs), and a suite of learning algorithms. Meta-AQUA then retrieves a simple XP of the form $\alpha \rightarrow \beta$ such that the antecedent (alpha) is hunger and the consequent (beta) is screaming. That is, the Wumpus screams, because it is hungry (XP-WUMPUS-SCREAMS.3432).

Now the system attempts to resolve the situation. Usually Meta-AQUA will seek a change of its knowledge to compensate for an apparent anomaly in a situation. The assumption is that the observer is passive. INTRO entertains instead that a goal can be spawned to resolve the anomaly by planning and executing actions that remove the antecedent of the XP. Once the antecedent is gone, the screaming will cease. Thus the resulting goal is to remove the hunger, and the goal is passed to the Prodigy/Agent component.

The algorithm is defined more formally in table 4. Cox and Ram (1999a) provide technical details related to steps 1 and 2.

Step 3 determines the substitution set with which to instantiate the explanation of the event. Step 4 checks the instantiated preconditions of the explanation. If the preconditions hold, the negation of the explanation's antecedent is posted as a goal.

Although this example is extremely simple and rather contrived, more realistic and detailed examples exist. In general, an XP con-

tains a set of antecedents (rather than the single antecedent α) called the XP asserted nodes (Ram 1993). Each of them must be true for the explains node (the event being explained) to hold. If any are removed by achieving their negation, then the causal structure will no longer hold. So in principle our simple Wumpus example can generalize to more complex behavior. Whether it scales well is another (future) issue. However, using this mechanism, Meta-AQUA has successfully processed thousands of short randomly generated stories (Cox 1996a) similar to the hundred stories from figure 4.

The Prodigy/Agent Planning Component

Prodigy/Agent¹⁰ (Cox et al. 2001, Elahi and Cox 2003) is an independent state-space planning agent that uses a predefined communication protocol represented in KQML to accept planning requests and to return a sequence of actions that achieve the planning goals. It is built around the PRODIGY planning and learning architecture described in the "Awareness and PRODIGY" section.

Planners are traditionally given specific goals to achieve by generating a sequence of actions that alters the physical environment. Yet a perpetual agent should be able to generate its own goals. We as humans have expectations about the world, how it should behave, and how we like it. When we detect something anomalous that violates these expectations, we attempt to explain the situation to make sense of the situation. Given a satisfactory explanation, the

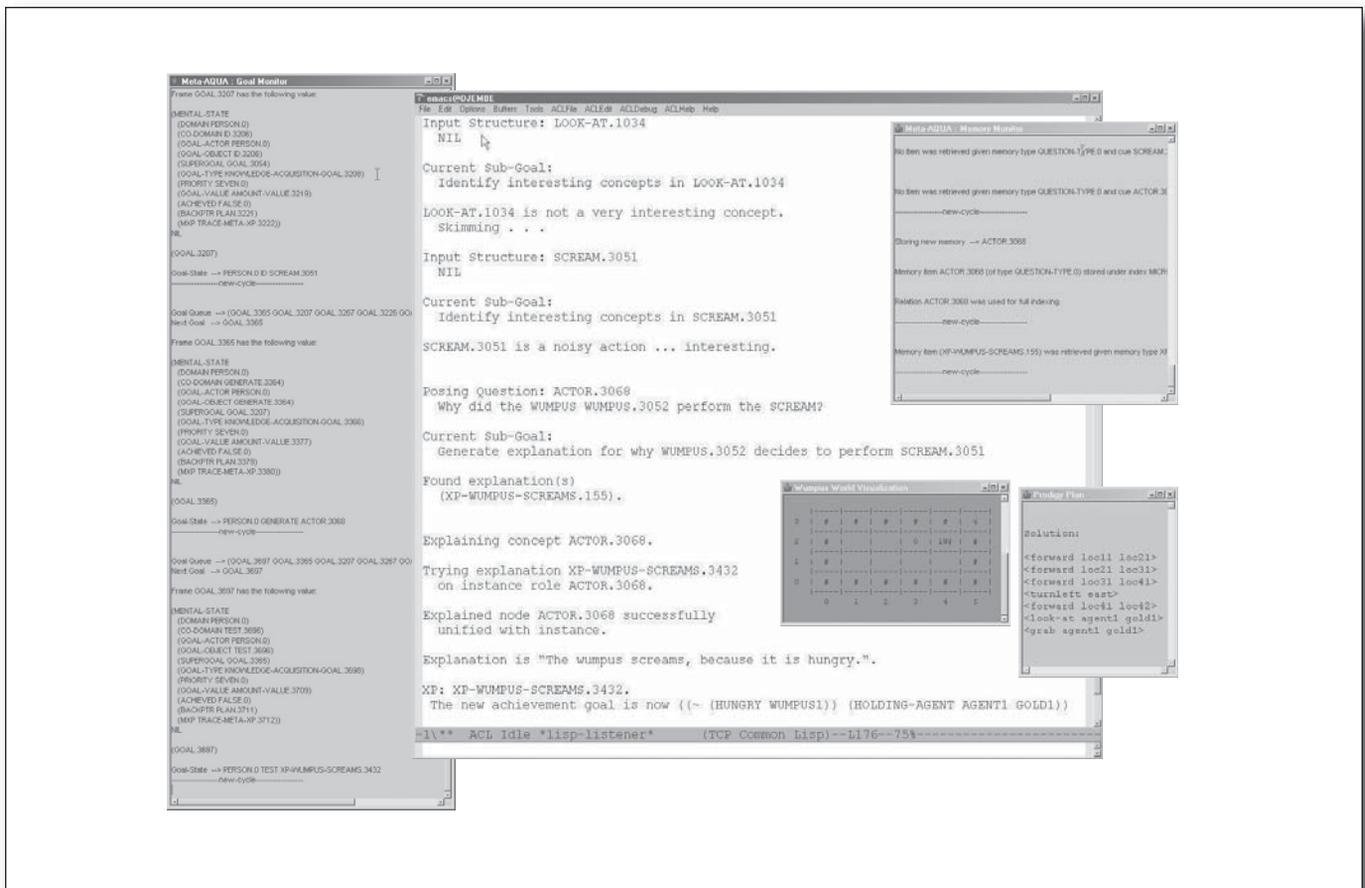


Figure 8. INTRO Output and Interface.

anomaly will be resolved. Hence we have an opportunity to learn something new about the world. Similar situations should no longer appear to be anomalous for us in the future. Given an unsatisfactory explanation, however, we may determine that something needs to be done to make the situation more to our liking. The result is a self-determined planning goal.

So given the goal to remove the hunger state and the initial state of the world by the Meta-AQUA module, Prodigy/Agent generates a new plan containing the action of feeding the Wumpus. When this plan is executed, no anomalous event occurs, because the reason for the unexpected behavior is no longer present in the environment. That is, in response to an active environment, INTRO generates its own goals to change the world.

Now to be clear, INTRO begins this task by providing Prodigy/Agent the standard goal to possess the gold, and this goal is given without appeal to experience. PRODIGY then creates a plan to enter the hex with the gold, look for it, and grab it (thereby possessing it). However when the agent enters the cell with the gold, the Wumpus is there also. Because the agent is

in the same cell and the Wumpus is hungry, it screams. This event then triggers the explanation, which leads to the new goal of not having the Wumpus hungry and to feeding the Wumpus and stopping the screaming. The INTRO program then halts. Nevertheless this sequence outlines a mechanism that relaxes the dependency upon user-supplied goals; it starts to provide a locus of inquiry regarding the origin of independent planning goals and offers up a knowledge-level alternative to the blind maximization of value functions and the programmer's burden of supplying all possible background goals.

The agent enters the location and can simply grab the gold to achieve its given goal. The Wumpus is benign and does not pose an actual threat to the goal, but something is missing in this interpretation. The screaming demands attention and a response. But why? Why do we stop to help a stranger in need or stoop to pick up a piece of garbage along the curb when we recognize such problems in an otherwise unrelated context? Our implementation starts to answer these questions, but just barely. A rational person would not decide to stop loud

1. Detect interesting event, β'
2. Retrieve an explanation, $E: \alpha \rightarrow \beta$, that covers anomaly
3. $\sigma \leftarrow \text{Unify}(\beta, \beta')$
4. Verify $\text{Subst}(\sigma, \text{precond}(E))$
5. Post goal, $G = \text{Achieve}(\neg \text{Subst}(\sigma, \alpha))$

Table 4. Explanation-Based Goal Formulation.

traffic noises by putting sugar in gas tanks, given the explanation that car engines make noise only when having clean fuel. This is the case despite the fact that if sugar is in all of the gas tanks, the fuel is not clean, and the noise will cease. Such a strategy is impractical and has undesirable consequences. Yet one might call the police on a single loud muffler. But note also that the choice of notifying the police has nothing to do with the direct mechanical explanation of why the car is making the noise. The police choice follows from an explanation of social convention, volitional decision making, legal responsibilities, and agency.

In the limit, we wish for an agent that can *tinker*. It must be able to push on and explore the boundaries of its own knowledge. An agent needs to discover some of the interactions between actions and animate objects, the causal connections between agents in a social network, and the relationship between the physical and mental environments in which it exists to be able to effectively explain the observations it encounters and to create new goals or to change existing goals¹¹ as a result.

Between Learning Goals and Achievement Goals

One of the most significant outstanding issues involved with the research presented here relates to distinguishing the requirements associated with learning goals and with achievement goals. When a system understands that its knowledge is flawed, it needs to generate a goal to change its own knowledge so that it is less likely to repeat the reasoning error that uncovered the flaw. When the world is "flawed," a system needs to generate a goal to achieve an alternative state of the world. The main issue is detecting the conditions under which a system does the latter versus the former. How does INTRO know that its knowledge

of screaming is not the problem that needs to be resolved in the Wumpus World scenario?

Currently the system is simply hard-coded to automatically generate achievement goals and then to plan for them. Future research remains to implement a sufficient decision process to differentiate the two goal types. But consider that clues will exist in the context provided by the series of events experienced and the introspective trace of the reasoning that accompanies action and deliberation in the environment. Besides it is stretching our imagination to entertain that somehow modifying our definition of a screaming event will result in the Wumpus not screaming in the future. Wishing that the Wumpus not be loud does not make it so.

Moorman (1997; Moorman and Ram 1999) presents a theory of reading for science fiction and similar stories that require the willing suspension of (dis)belief. The theory presents a matrix of conceptual dimensions along which a knowledge structure can be moved to analogically map between inputs. Thus to view a robot as human is a smaller shift in the matrix than is to view a robot somehow as a space-time event. Understanding science fiction requires such conceptual shifts, even though the reasoner may not believe that a robot is truly human (a living organism).

When understanding events and objects in any environment (fictional or not), judgments as to the reasonableness of possibilities do exist in the natural world. Thus it is rational to consider feeding the Wumpus, because an action actually exists to achieve the goal; whereas the alternative is too strange. A system might also make the decision to generate an achievement goal over a learning goal based upon its experience with general screaming events and the relative certainty of such knowledge. Note that to do so, a system must evaluate its own knowledge, experience, and capability; it must use or create metaknowledge.

Problems exist with these speculations, however. For example, basing a decision upon the fact that the system can execute a plan to feed the Wumpus requires that the system reason about the structure of a plan before the plan is computed. Similarly to reason about the potential learning goal (to change the concept of screaming) requires the system to consider steps in a learning plan before the system can perform the planning. In either case the solution is not to be found solely within the traditional automated planning community nor within the knowledge representation community. Rather the capability to determine the category of goal worth pursuing is tightly coupled with metacognitive activity.

Self-Awareness

A renewed interest exists in machines that have a metacognitive or introspective capacity, implement metareasoning, incorporate meta-knowledge, or are otherwise in some way self-aware (see Cox [2005] for a thorough review). Yet little consensus exists in the AI community as to the meaning and use of such mental terms, especially that of self-awareness. But before considering what it might mean for a machine to be self-aware, consider what it means to be aware at all. A weak sense of the word does exist. For example your supervisor may tell you that "I am *aware* of your problem." Here awareness appears to be less empathetic than in the statement "I *understand* your problem." In the first sense awareness is simply a registration of some state or event, ignoring for the moment consciousness. Awareness need not be magical.

Consider then what it means to be aware in the sense of understanding the world. To understand it is not simply to classify objects in the environment into disjunct categories. Rather it is to interpret the world with respect to the knowledge and experience one (human or machine) currently has in memory. The case-based reasoning community suggests that it is to find a piece of knowledge, schema, or case most relevant to its conceptual meaning and to apply it to the current situation so that a new structure can be built that provides causal linkages between what has already occurred and what is likely to occur next; that is, it provides causal explanation and expectation (Kolodner 1993; Leake 1992; Ram 1993; Schank 1986; Schank, Kass, and Riesbeck 1994). Understanding or awareness is not just perceiving the environment. It is certainly not logical interpretation as a mapping from system symbols to corresponding objects, rela-

tions, and functions in the environment. Here I claim that acute awareness of the world implies being able to comprehend when the world is in need of change and, as a result, being able to form an independent goal to change it.

Likewise, being self-aware is not just perceiving the self in the environment, nor is it simply possessing information about the self; rather it is *self-interpretation* (see also Rosenthal [2000]). It is understanding the self well enough to generate a set of explicit learning goals that act as a target for improving the knowledge used to make decisions in the world. It is also understanding the self well enough to explain ourselves to others (for example, see Johnson [1994]; Core et al. [2006]; van Lent, Fisher, and Mancuso [2004]). To equate self-awareness with conscious direct experience is missing the point. Many nonconscious correlates such as implicit memory are highly associated with self-awareness and metacognition (Reder and Schunn 1996). Some (for example, at the DARPA Workshop on Self-Aware Computer Systems [McCarthy and Chaudri 2004]) have suggested that self-aware systems are linked in a special way with metacognition. But if one follows a straightforward definition of metacognition as cognition about cognition, then representing a trace of reasoning and then reasoning about the trace is sufficient. Prodigy/Analogy does represent the rationale for its planning decisions and can reason about the rationale when applying further planning. Yet the program has no reference to itself other than the annotations of justifications on its search tree nodes. Meta-AQUA represents goals as relations between a volitional agent (itself) and the state it desires. Yet it never relies upon the symbol for itself in any of its processing at the base level or metalevel. However, to reason about the self without an explicit structural representation of the self seems less than satisfactory.

Thus, as currently implemented, INTRO is just that, an introduction. What is required is a more thorough integration of the metacognitive knowledge and processes in a system like INTRO. This article and the INTRO implementation have concentrated on a cognitive integration of planning, comprehension, and learning; a metacognitive integration between Meta-AQUA and PRODIGY remains unfinished.

Acknowledgments

I thank Sylvia George for the help in implementing the translation code and for the modifications to the Wumpus World code. She also

wrote much of the PRODIGY domain specification for the Wumpus World problem and contributed some of the INTRO-specific frame representations to Meta-AQUA so that it could process and explain the scream event. Finally I thank Mark Burstein, David McDonald, Kerry Moffitt, John Ostwald, and the anonymous reviewers for their helpful comments.

Notes

1. But see also Singh (2005) and Minsky, Singh, and Sloman (2004) for a complementary approach. Brachman (2002) has also called for a greater emphasis upon the integration of cognitive and metacognitive capabilities.
2. Cox (1996b, pp. 294–299) discusses some of the intersections between problem solving and comprehension represented in the letter C region in the top center of the figure. For example, a problem solver must be able to monitor the execution of a solution to confirm that it achieves its goal. If the comprehension process determines that the goal pursuit is not proceeding as desired, then the execution failure must be addressed and the solution changed.
3. The Meta-AQUA home page is at meta-aqua.mcox.org. Version 6 of the code release, related publications, and support documents exist there.
4. The University of Maryland Nonlin home page exists at www.cs.umd.edu/projects/plus/Nonlin/.
5. The PRODIGY home page exists at www.cs.cmu.edu/afs/cs.cmu.edu/project/prodigy/Web/prodigy-home.html.
6. The INTRO home page exists at meta-aqua.mcox.org/intro.html.
7. The code we modified is located at aima.cs.berkeley.edu/code.html. See the acknowledgments.
8. The system also finds all anomalies that diverge from expectations to be interesting as well as concepts about which it has recently learned something. In any case the situation is surprising, and this constitutes sufficient grounds for being interesting and in need of explanation.
9. In the main INTRO window, the goal is shown as the frame ACTOR.3068. The actor frame represents the relation between the action and the agent who did the action. That is, it is the relation facet of the actor slot whose value facet is the Wumpus. The explanation (that is, answer to the question) is a representation of why the Wumpus “decided” to perform the scream event.

10. See www.mcox.org/Prodigy-Agent/ for a public version of the implemented system, the user manual, and further details.

11. See Cox and Veloso (1998) and Cox and Zhang (2005) for a theory of goal change in planning

References

- Brachman, R. J. 2002. Systems That Know What They Are Doing. *IEEE Intelligent Systems* 17(6)(November–December): 67–71.
- Carbonell, J. G. 1986. Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In *Machine Learning: an Artificial Intelligence Approach: Volume 2*, ed. R. Michalski, J. Carbonell, and T. M. Mitchell, 371–392. San Francisco: Morgan Kaufmann Publishers.
- Carbonell, J. G.; Blythe, J.; Etzioni, O.; Gil, Y.; Joseph, R.; Kahn, D.; Knoblock, C.; Minton, S.; Perez, A.; Reilly, S.; Veloso, M.; and Wang, X. 1992. PRODIGY 4.0: The Manual and Tutorial, Technical Report, CMU-CS-92-150, Computer Science Dept., Carnegie Mellon University, Pittsburgh, PA.
- Carbonell, J. G.; Knoblock, C. A.; and Minton, S. 1991. PRODIGY: An Integrated Architecture for Planning and Learning. In *Architectures for intelligence: The 22nd Carnegie Mellon Symposium on Cognition*, ed. K. Van Lehn, 241–278. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Chien, S.; Hill, Jr., R.; Wang, X.; Estlin, T.; Fayyad, K.; and Mortenson, H. 1996. Why Real-World Planning Is Difficult: A Tale of Two Applications. In *New Direction in AI Planning*, ed. M. Ghallab and A. Milani, 287–298. Amsterdam: IOS Press.
- Core, M. G.; Lane, H. C.; van Lent, M.; Gomboc, D.; Solomon, S.; and Rosenberg, M. 2006. Building Explainable Artificial Intelligence Systems. In *Proceedings of the Eighteenth Conference on Innovative Applications of Artificial Intelligence*. Menlo Park, CA: AAAI Press.
- Cox, M. T. 2005. Metacognition in Computation: A Selected Research Review. *Artificial Intelligence* 169(2): 104–141.
- Cox, M. T. 1997. An Explicit Representation of Reasoning Failures. In *Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning*, ed. D. B. Leake and E. Plaza, 211–222. Berlin: Springer-Verlag.
- Cox, M. T. 1996a. An Empirical Study of Computational Introspection: Evaluating Introspective Multistrategy Learning in the Meta-AQUA System. In *Proceedings of the Third International Workshop on Multistrategy Learning*, ed. R. S. Michalski and J. Wnek, 135–146. Menlo Park, CA: AAAI Press/The MIT Press.
- Cox, M. T. 1996b. Introspective Multistrategy Learning: Constructing a Learning Strategy under Reasoning Failure, Technical Report, GIT-CC-96-06. Ph.D. dissertation, College of Computing, Georgia Institute of Technology, Atlanta, GA (hcs.bbn.com/personnel/Cox/thesis/).
- Cox, M. T. 1994a. Case-based Introspection. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1435. Menlo Park, CA: AAAI Press.
- Cox, M. T. 1994b. Machines That Forget: Learning from Retrieval Failure of Mis-indexed Explanations. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, 225–230. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Cox, M. T.; Edwin, G.; Balasubramanian, K.; and Elahi, M. 2001. Multiagent Goal Transformation and Mixed-Initiative Planning Using Prodigy/Agent. In *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics, Vol. VII*, ed. N. Callaos, B. Sanchez, L. H. Encinas, and J. G. Busse, 1–6. Orlando, FL: International Institute of Informatics and Systemics.
- Cox, M. T.; Muñoz-Avila, H.; and Bergmann, R. 2006. Case-based Planning. *Knowledge Engineering Review* 20(3): 283–287.
- Cox, M. T., and Ram, A. 1999a. Introspective Multistrategy Learning: On the Construction of Learning Strategies. *Artificial Intelligence* 112(1–2): 1–55.
- Cox, M. T., and Ram, A. 1999b. On the Intersection of Story Understanding and Learning. In *Understanding Language Understanding: Computational Models of Reading*, ed. A. Ram and K. Moorman, 397–434. Cambridge, MA: The MIT Press/Bradford Books.
- Cox, M. T., and Ram, A. 1995. Interacting Learning-Goals: Treating Learning as a Planning Task. In *Advances in Case-based Reasoning*, ed. J.-P. Haton, M. Keane, and M. Manago, 60–74. Berlin: Springer.
- Cox, M. T., and Veloso, M. M. 1998. Goal Transformations in Continuous Planning. Paper presented at the 1998 AAAI Fall Symposium on Distributed Continual Planning, Orlando, FL.
- Cox, M. T., and Veloso, M. M. 1997. Supporting Combined Human and Machine Planning: An Interface for Planning by Analogical Reasoning. In *Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning*, ed. D. B. Leake and E. Plaza, 531–540. Berlin: Springer-Verlag.
- Cox, M. T., and Zhang, C. 2005. Planning as Mixed-initiative Goal Manipulation. In *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling*, ed. S. Biundo, K. Myers, and K. Rajan,

- 282–291. Menlo Park, CA: AAAI Press.
- Elahi, M., and Cox, M. T. 2003. User's Manual for Prodigy/Agent, Ver. 1.0, Technical Report, WSU-CS-03-02, Department of Computer Science and Engineering, Wright State University, Dayton, OH.
- Fox, S. 1996. Introspective Learning for Case-Based Planning, Technical Report, 462. Ph.D. dissertation, Indiana University, Computer Science Dept., Bloomington, IN.
- Fox, S.; and Leake, D. B. 1995. Learning to Refine Indexing by Introspective Reasoning. In *Proceedings of the First International Conference on Case-Based Reasoning*, Lecture Notes in Computer Science 1010, 430–440. Berlin: Springer Verlag.
- Ghosh, S.; Hendler, J.; Kambhampati, S.; and Kettler, B. 1992. The UM Nonlin Planning System [a Common Lisp Implementation of A. Tate's Nonlin Planner]. College Park, MD: University of Maryland (ftp://cs.umd.edu/pub/nonlin/nonlin-files.tar.Z).
- Haigh, K. Z.; Kiff, L. M.; and Ho, G. 2006. The Independent LifeStyle Assistant™ (I.L.S.A.): Lessons Learned. *Assistive Technology* 18: 87–106.
- Johnson, W. L. 1994. Agents That Learn to Explain Themselves. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI.
- Kolodner, J. L. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.
- Leake, D. B. 1992. *Evaluating Explanations: A Content Theory*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Lee, P., and Cox, M. T. 2002. Dimensional Indexing for Targeted Case-Base Retrieval: The SMIRKS System. In *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*, ed. S. Haller and G. Simmons, 62–66. Menlo Park, CA: AAAI Press.
- McCarthy, J., and Chaudri, V. 2004. DARPA Workshop on Self Aware Computer Systems, 27–28 April. Arlington, VA: SRI Headquarters.
- Minsky, M.; Singh, P.; and Sloman, A. 2004. The St. Thomas Common Sense Symposium: Designing Architectures for Human-Level Intelligence. *AI Magazine* 25(2): 113–124.
- Moorman, K. 1997. A Functional Theory of Creative Reading: Process, Knowledge, and Evaluation, Ph.D. dissertation, College of Computing, Georgia Institute of Technology, Atlanta, GA.
- Moorman, K., and Ram, A. 1999. Creativity in Reading: Understanding Novel Concepts. In *Understanding Language Understanding: Computational Models of Reading*, ed. A. Ram and K. Moorman, 359–395. Cambridge, MA: The MIT Press/Bradford Books.
- Muñoz-Avila, H., and Cox, M. T. 2006. *Case-based Plan Adaptation: An Analysis and Review*. Unpublished (www.cse.lehigh.edu/~munoz/Publications/MunozCox06.pdf).
- Murdock, J. W. 2001. Self-Improvement through Self-understanding: Model-based Reflection for Agent Adaptation. Ph.D. dissertation, College of Computing, Georgia Institute of Technology, Atlanta, GA.
- Murdock, J. W., and Goel, A. K. 2001. Meta-Case-Based Reasoning: Using Functional Models to Adapt Case-based Agents. In *Case-Based Reasoning Research and Development: Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR-2001*, ed. D. W. Aha, I. Watson, and Q. Yang, 407–421. Berlin: Springer.
- Myers, K., and Yorke-Smith, N. 2005. A Cognitive Framework for Delegation to an Assistive User Agent. In *Mixed-Initiative Problem-Solving Assistants: Papers from the 2005 AAAI Fall Symposium*, AAAI Technical Report FS-05-07. Menlo Park, CA: American Association for Artificial Intelligence.
- Pollack, M. E., and Horty, J. F. 1999. There's More to Life than Making Plans. *AI Magazine* 20(4): 71–83.
- Ram, A. 1993. Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases. *Machine Learning* 10: 201–248.
- Ram, A. 1991. A Theory of Questions and Question Asking. *The Journal of the Learning Sciences* 1(3–4): 273–318.
- Ram, A., and Leake, D., eds. 1995. *Goal-Driven Learning*. Cambridge, MA: MIT Press/Bradford Books.
- Reder, L. M., and Schunn, C. D. 1996. Metacognition Does Not Imply Awareness: Strategy Choice Is Governed by Implicit Learning and Memory. In *Implicit Memory and Metacognition*, ed. L. Reder, 45–77. Mahwah, NJ: Lawrence Erlbaum Associates.
- Rosenthal, D. 2000. Introspection and Self-Interpretation, *Philosophical Topic* (special issue on introspection) 28(2): 201–233.
- Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*, 2nd. ed. Upper Saddle River, NJ: Prentice Hall.
- Schank, R. C. 1979. Interestingness: Controlling Inferences. *Artificial Intelligence* 12(3): 273–297.
- Schank, R. C. 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge, MA: Cambridge University Press.
- Schank, R. C. 1986. *Explanation Patterns: Understanding Mechanically and Creatively*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Schank, R. C.; Kass, A.; and Riesbeck, C. K. 1994. *Inside Case-Based Explanation*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Singh, P. 2005. EM-ONE: An Architecture for Reflective Commonsense Thinking. Ph.D. dissertation. Department of Electrical Engineering and Computer Science. Massachusetts Institute of Technology, Cambridge, MA.
- Tate, A. 1976. Project Planning Using a Hierarchic Non-linear Planner (Tech. Rep. No. 25). Edinburgh, UK: University of Edinburgh, Department of Artificial Intelligence.
- van Lent, M.; Fisher, W.; Mancuso, M. 2004. An Explainable Artificial Intelligence System for Small-Unit Tactical Behavior. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 900–907. Menlo Park, CA: AAAI Press.
- Veloso, M. M. 1994. *Planning and Learning by Analogical Reasoning*. Berlin: Springer.
- Veloso, M., and Carbonell, J. G. 1994. Case-based Reasoning in PRODIGY. In *Machine Learning IV: A Multistrategy Approach*, ed. R. S. Michalski and G. Tecuci, 523–548. San Francisco: Morgan Kaufmann Publishers.
- Veloso, M.; Carbonell, J. G.; Perez, A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Theoretical and Experimental Artificial Intelligence* 7(1): 81–120.



Michael T. Cox is a senior scientist in the Intelligent Distributing Computing Department of BBN Technologies, Cambridge, MA. Previous to this position, Cox was an assistant professor in the Department of Computer Science and Engineering at Wright State University, Dayton, Ohio, where he was the director of Wright State's Collaboration and Cognition Laboratory. His research interests include case-based reasoning, collaborative mixed-initiative planning, understanding (situation assessment), introspection, and learning. More specifically, he is interested in how goals interact with and influence these broader cognitive processes.

New & Recent Books from AAI Press

Twenty-First National Conference on Artificial Intelligence

Yolanda Gil and Raymond Mooney,
Program Chairs
NOW SHIPPING!

Data Mining: Next Generation Challenges and Future Directions

Hillol Kargupta, Anupam Joshi,
Krishnamoorthy Sivakumar, and Yelena Yesha
www.aaai.org/Press/Books/kargupta2.php

International Conference on Automated Planning and Scheduling 2006

Derek Long and Stephen Smith,
Program Chairs
NOW SHIPPING!

New Directions in Question Answering

Mark Maybury
www.aaai.org/Press/Books/maybury3.php

Nineteenth International FLAIRS Conference

Geoff Sutcliffe and Randy Goebel,
Program Chairs
NOW SHIPPING!

Tenth International Conference on Principles of Knowledge Representation and Reasoning

Patrick Doherty and John Mylopoulos,
Program Chairs
NOW SHIPPING!

Thinking about Android Epistemology

Kenneth Ford, Clark Glymour,
and Patrick J. Hayes
www.aaai.org/Press/Books/ford.php

Smart Machines in Education

Kenneth Forbus
and Paul Feltovich
www.aaai.org/Press/Books/forbus.php

Editor-in-Chief

Anthony Cohn, *University of Leeds, UK*

AAAI Press Editorial Board

Aaron Bobick,
Georgia Institute of Technology, USA
Ken Forbus,
Northwestern University, USA
Enrico Franconi,
Free University of Bozen - Bolzano, Italy
Thomas Hofmann,
Darmstadt University of Technology, Germany
Craig Knoblock,
Information Sciences Institute,
University of Southern California, USA
George Luger,
University of New Mexico, USA
David Poole,
University of British Columbia, Canada
Oliviero Stock,
ITC IRST, Italy
Gerhard Widmer,
Johannes Kepler University, Austria
Mary Anne Williams,
University of Technology, Sydney, Australia

Editor-in-Chief Emeritus

Kenneth Ford, *Institute for*
Human and Machine Cognition, USA