Perpetual Self-Aware Cognitive Agents

Michael T. Cox

BBN Technologies
10 Moulton St.
Cambridge, MA 02138
mcox@bbn.com

Abstract

To construct a *perpetual self-aware cognitive agent* that can continuously operate with independence, an introspective machine must be produced. To assemble such a 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. As a start toward this goal, I performed an integration of an introspective multistrategy learning system with a nonlinear state-space planning agent using the wumpus world as environment. In this integration I show how the resultant system I call INTRO can generate its own goals. I use this system to discuss issues of self-awareness by machine.

Introduction

Although by definition all AI systems can perform intelligent activities, virtually none can understand why they do what they do, nor how. Moreover without metacognition, a system can do intelligent actions without knowing that it is doing intelligent actions, let alone why. 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, meta-level architecture. The AI community currently recognizes that first-order reasoning in isolation (e.g., puzzle problem-solving) is insufficient, and therefore a full situated agent (i.e., an adaptive problem-solver possessing perception of and action within a complex environment) is necessary. Likewise, a second-order metacognitive 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 paper examines a comprehensive approach to the production of an agent that understands itself and the world around it in a meaningful way. Rather than just a machine specialized to operate within a complex physical world, this machine must also be able to understand itself within such a context.[1]

Figure 1 shows a decomposition of the relationships between 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 planning task. Cox and Ram (1995) discuss this analogy at length.[2] Cox and Ram (1999b) examines the similarity between learning and story understanding as indicated by the intersection labeled B on the right in Figure 1. Here I discuss in some detail issues related to the letter D intersection (letter C is in the footnote).

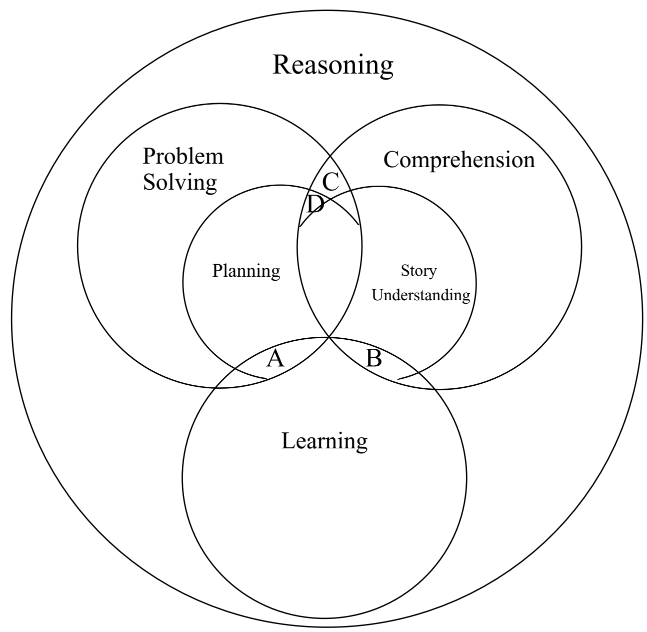

Figure 1: Hierarchical decomposition of reasoning

Section D represents the intersection of planning and comprehension, normally studied separately. Planning is more than generating a sequence of actions that if executed with transform some initial state into a given goal state.

1.But see also Minsky, Singh, and Sloman (2004).

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 planned, then the planning failure must be addressed and the plan changed.

Instead planning is embedded in a larger plan management process that must interleave planning, execution, and plan understanding (Chien, Hill, Wang, Estlin, Fayyad and Mortenson 1996; Pollack and Horty 1999). Furthermore all AI systems accept goals as input or have an inherent background goal that drives system behavior; no system derives its own explicit goals given an understanding (i.e., comprehension) of the environment other than through reactive means.

The major objective of this paper is to show how a preliminary system called INTRO can systematically create 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 it uses for explanation is the same that its metareasoning component uses when explaining a failure of base-level reasoning. The resulting goals in the latter case seek to change its knowledge in order to reduce the chance of repeating the reasoning failure. (i.e., a learning goal is to change the dissonance between what it knows and what it should know).

INTRO

This section describes a preliminary implementation of an INitial inTROspective cognitive-agent called INTRO that is designed to exist continually in a given environment. It represents an example of life-long learning agent and a perpetual agent that can generate explicit declarative goals that provide deliberate intention and a focus for activities (see Ram and Leake 1995 for a functional argumentation for the role of explicit goals).

The agent itself has four major components (see Figure 2). INTRO has primitive perceptual and effector subsystems and has two more sophisticated cognitive subsystems. The latter two compose both the planning and understanding components and consist of the Prodigy/Agent and Meta-AQUA systems respectively.

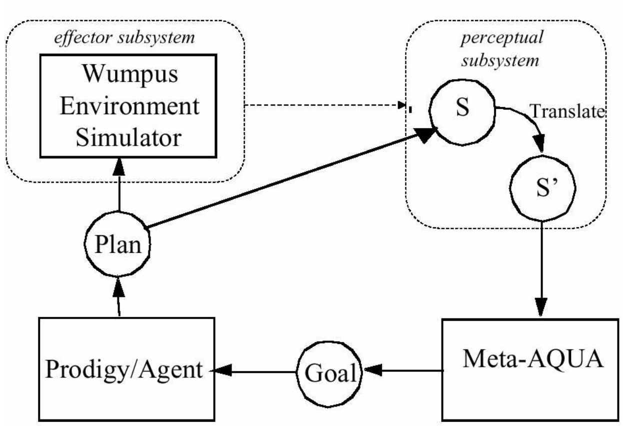

Figure 2: INTRO architecture

The Wumpus World

A common agent environment (especially for pedagogical purposes) exists publicly as the Wumpus World[3] (Russell and Norvig 2003). This environment contains an agent whose goal is to find the gold while avoiding pits and the wumpus creature. Unlike the Wumpus, the agent can perform actions to change the environment such as to turn, move ahead, pickup, 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 a 5-tuple that represents whether the Wumpus is nearby, the Wumpus screams, a pit is nearby, gold is co-located, and whether an obstacle has been encountered.

For the purposes of this paper, the environment has been limited to a four by one cell world (i.e., a length-4 corridor). The agent as always starts in cell [1,1] at the western-most end of the corridor. The Wumpus and the gold can be placed in any of the remaining three cells. The example I use here places the Wumpus and gold at the eastern-most end of the corridor in every initial state (see Figure 3). In a major change to the interpretation of the game, the Wumpus does not scream as it dies from an accurate arrow. The Wumpus is rather benign in our example, because it will not injure the agent. It screams instead, because it is hungry. Furthermore the agent can choose the action to feed the Wumpus. This will prevent screaming.

```
   |----|----|----|----|----|----|
 2 | #  | #  | #  | #  | #  | #  |
   |----|----|----|----|----|----|
 1 | #  | 1> |    |    | W$ | #  |
   |----|----|----|----|----|----|
 0 | #  | #  | #  | #  | #  | #  |
   |----|----|----|----|----|----|
     0    1    2    3    4    5
```

Figure 3: Initial Wumpus World state

Normally the agent control program maps an input percept to an output action choice given a current percept and any knowledge it possesses. We have modified the agent control code to accept as input a series of actions from plans output by the Prodigy/Agent component of INTRO. The simulator then presents a visualization of the events. As the implementation currently stands, the output of the simulator is not used. Ideally (as shown in the dashed arrow of Figure 2) the output should be input into the perceptual component.

The Perceptual Subsystem

The perceptual subsystem [*sic*] does not really present a realistic perception or filtering of the world and its actions. Instead the module in its present form acts as a translator between the representation of Prodigy/Agent and Meta-AQUA. Prodigy/Agent uses a STRIPS-like operator repre-

3. The code we modified is the Russell and Norvig program at the URL `aima.cs.berkeley.edu/code.html`. See acknowledgments.

sentation language whose BNF is provided in Carbonell, Blythe, Etzioni, Gil, Joseph, Kahn, Knoblock, Minton, Perez, Reilly, Veloso, and Wang (1992). Meta-AQUA uses representations implemented as frames (Cox 1997). The representation is based upon the XP-theory of Schank (1986; Schank, Kass, and Riesbeck 1994) and includes conceptual dependency primitives and memory organization packets or MOPs (Schank 1982).

The problem of translation is one of mapping a flat STRIPS operator to an arbitrarily deep, hierarchical, slot-filler case event. For example the action SCREAM (wumpus1 loc4 gold1 agent1) must translate into the scream frame representation as illustrated in Figure 4. Problems exist when the parameters do not match in terms of both names and content and when they differ in relative order. To resolve such problems, I wrote a mapping function for translation.

The function call (translate ‘SCREAM ‘(wumpus1 loc4 gold1 agent1) ‘(1 (3 2) 0 2)) resolves the mismatch in representations between the planning operator and conceptual interpretation frame. Thus the object wumpus1 maps to the `actor` slot, agent1 maps to the `object` slot, the gold1 object is ignored, and loc4 is set to the `co-domain` of the `at-location` frame in the `to` slot. This method also assumes a linked correspondence between the two `scream` symbols. They happen to be the same in this case, but in general they may be completely different.

The Meta-AQUA System

Meta-AQUA (Cox 1996b; Cox and Ram 1999a; Lee and Cox 2002) is an introspective multistrategy learning system that improves its story understanding performance task through a metacognitive analysis of reasoning failures. Understanding the actions of a story and the reasons why actors perform such actions is very similar to comprehending the actions and motivations of agents in an environment. So within the INTRO cognitive agent, Meta-AQUA does perform this task. 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.

As is shown in the example INTRO output of Figure 5, Meta-AQUA inputs three forward movements by the Wumpus that are not interesting in any significant way. These actions (displayed in the Prodigy/Agent plan window and in the main INTRO activity window within the emacs Common Lisp buffer) are therefore skimmed and simply inserted into the model of the wumpus world actions. However when the system encounters the scream action it is processed differently, because sex, violence, and loud noises are inherently interesting (Schank 1979).[4] Notice that Meta-AQUA presents two special windows that display internal representations of the cognitive processing. In the “Goal Monitor” window, the system shows a goal to identify interesting events in the input. The scream input causes Meta-AQUA to spawn a new goal to generate an explanation for the scream. This goal is a knowledge goal or question whose answer explains why the Wumpus performed the action.[5] The “Memory Monitor” window display a representation of the memory retrieval and storage activities along with the indexes and mental objects that occupy memory.

```
(OPERATOR SCREAM
  (params <wumpus> <location1> <gold> <agent>)
  (preconds
   ((<wumpus> WUMPUS)
    (<location1> LOCATION)
    (<gold> GOLD)
    (<agent> AGENT))
   (and (at-wumpus <wumpus> <location1>)
        (at-agent <agent> <location1>)))
  (effects
   ()
   ((add (dropped <gold>))
    (del (quiet-wumpus <wumpus>))
    (del (holding-wumpus <wumpus> <gold>))
    (add (screaming-wumpus <wumpus>)))))

(define-frame SCREAM
  (isa (value (noisey-mop)))
  (actor (value (wumpus)))
  (object (value (animate-object)))
  (to (value (at-location
             (domain (value =object)))
             (co-domain (value loc-value))))
  (instrumental-scene
    (value (speak
     (actor (value =actor))
     (object (value (animal-noises)))
     )))
  (goal-scene
    (value
      (mtrans
        (actor (value =actor))
        (object (value (knowledge-state =ks
                       (domain (value =actor)))))
        (from (value (at-location
                       (domain (value =actor)))))
        (to (value (at-location
                   (domain (value =object))))))))
  (main-result (value =ks))
  (scenes (value (=instrumental-scene
                 =goal-scene))))
```

Figure 4: Two representations for a SCREAM action

4.The system also finds all anomalies interesting as well as concepts about which it has recently learned something.

5. In the main INTRO window, the goal is shown as the frame ACTOR.1096. 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 (i.e., answer to the question) is a representation of why the Wumpus “decided” to perform the scream event.

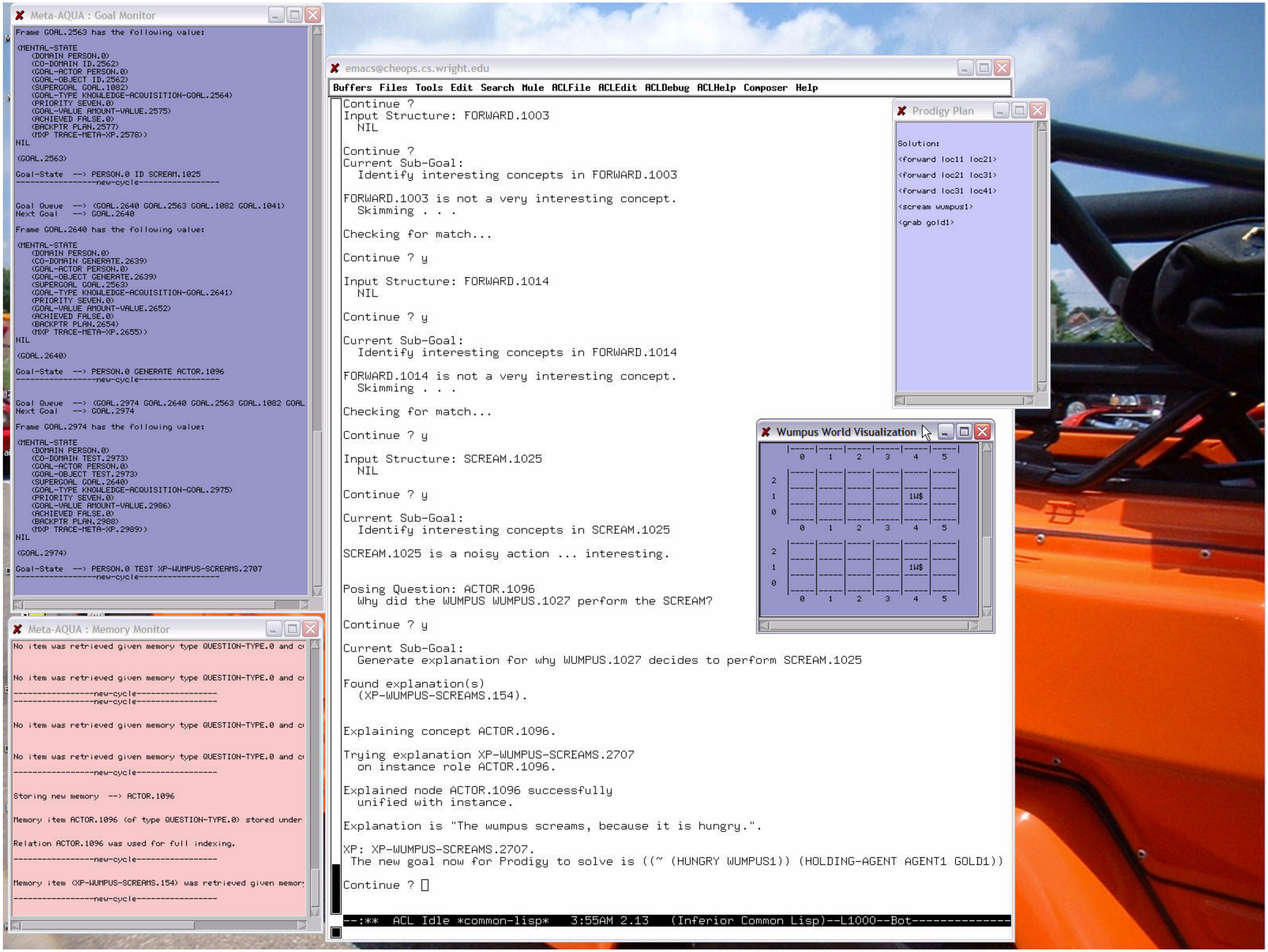

Figure 5: INTRO output and interface

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. Using case-based reasoning (see Cox and Ram 1999a, for the specific algorithm), Meta-AQUA retrieves a simple XP of the form $\alpha \to \beta$ such that the antecedent (alpha) is hunger and the consequent (beta) is screaming. That is the wumpus screams, because it is hungry.

Now the system attempts to resolve the unpleasant 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.

Although this example is extremely simply and rather contrived, more realistic and complex examples exist. In general an XP has a set of antecedents 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, then the causal structure will no longer hold. In effect our simple Wumpus example can generalize. 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).

The Prodigy/Agent System

Prodigy/Agent[6] (Cox, Edwin, Balasubramanian, and Elahi 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 (Veloso, Carbonell, Perez, Borrajo, Fink, and Blythe 1995). At its core is a nonlinear state-space planner called Prodigy 4.0 (Carbonell, *et al.* 1992). It follows a means-ends analysis backward-chaining search procedure that reasons about both multiple goals and multiple alternative operators from its domain theory.

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. Given a satisfactory explanation, 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.

Now 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.

Learning Goals or Achievement Goals

One of the most significant outstanding issues involved with the research presented here relates to the differentiation between 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 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 decision process. But consider that clues will exist in the context of the series of events and the 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 is not loud does not make it so.

Moorman (1997; Moorman and Ram 1995) presents a theory of reading for science fiction and other stories that require willful suspension of (dis)belief. The theory presents a matrix of conceptual dimensions along which a knowledge structure can be moved in order to analogize between inputs. Thus to view a robot as a man is a smaller shift in the matrix than is to view a robot somehow as an space-time event. Understanding science fiction requires such shifts, even though the reasoner may not believe that a robot is truly a man (a living organism).

When understanding events and objects in any environment (fictional or not), judgements 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

6. See `www.cs.wright.edu/~mcox/Prodigy-Agent` for a public versions of the implemented system, the user manual, and further details.

system must evaluate its own knowledge, experience, and capability; it must use or create metaknowledge.

Serious 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 likelihood of a plan before the plan is computed. Likewise 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 with the tradition automated planning community nor with the knowledge representation community. Rather metacognitive-like activity is tightly coupled with the capability to determine the kind of goal worth pursuing.

Self-Awareness

A renewed interest exists (e.g., McCarthy and Chaudhri 2004; see also Cox, 2005) in machines that have a metacognitive or introspective capacity, implement metareasoning, incorporate metaknowledge, or are otherwise self-aware. 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 exists. For example your supervisor may tell you that "I am *aware* of your problem." Here awareness appears to be less empathetic that 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.

Consider then what it means to be aware in the sense of understanding the world. To understand it is not to simply classify objects in the environment into disjunct categories. Rather it is to interpret it 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, relations 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 a 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. To equate self-awareness with conscious direct experience is missing the point. Many non-conscious correlates such as implicit memory are highly associated with self-awareness and metacognition (Reder and Schunn 1996). Some (for example at the DAPRA 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 we take a straightforward definition of metacognition as cognition about cognition, then representing a trace of reasoning and reasoning about the trace is sufficient. PRODIGY does represent the rationale for its planning decisions and can reason about the rationale when applying further planning (Veloso 1994). Yet PRODIGY 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 actually uses the symbol for itself in any of its processing at the base- or meta-level. However to reason about the self without an explicit representation of the self seems less than satisfactory. Thus as currently implemented, INTRO is just that, an introduction.

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 most of the planning domain for the wumpus world problem and added frame representations to Meta-AQUA so that it could process and explain the scream event. Finally I thank the anonymous reviewers for their helpful comments.

References

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.

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 M. Ghallab and A. Milani, eds. *New Direction in AI Planning*, 287-298. Amsterdam: IOS Press.

Cox, M. T. 2005. *Metacognition in Computation: A Selected History.* This volume.

Cox, M. T. 1997. An Explicit Representation of Reasoning Failures. In D. B. Leake and E. Plaza, eds., *Case-Based Reasoning Research and Development: Second International Conference on Case-Based Reasoning*, 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 R. S. Michalski and J. Wnek, eds., *Proceedings of the Third International Workshop on Multistrategy Learning*, 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. diss., College of Computing, Georgia Institute of Technology, ftp.cc.gatech.edu/pub/ai/ram/git-cc-96-06.html

Cox, M. T., Edwin, G., Balasubramanian, K., and Elahi, M. 2001. Multiagent Goal Transformation and Mixed-Initiative Planning Using Prodigy/Agent, in N. Callaos, B. Sanchez, L. H. Encinas, and J. G. Busse, eds., *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics, International Institute of Informatics and Systemics*: Orlando, FL, Vol. VII, 1-6.

Cox, M. T., and Ram, A. 1999a. Introspective Multistrategy Learning: On the Construction of Learning Strategies. *Artificial Intelligence, 112*: 1-55.

Cox, M. T., and Ram, A. 1999b. On the Intersection of Story Understanding and Learning. In A. Ram and K. Moorman, eds. *Understanding Language Understanding: Computational models of reading*, 397-434. Cambridge, MA: MIT Press/Bradford Books.

Cox, M. T., and Ram, A. 1995. Interacting Learning-Goals: Treating Learning as a Planning Task. In J.-P. Haton, M. Keane and M. Manago, eds. *Advances in case-based reasoning,* 60-74. Berlin: Springer-Verlag.

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.

Fox, S. 1996. *Introspective Learning for Case-Based Planning,* Technical Report, 462. Ph.D. diss., 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*.

Kolodner, J. L. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.

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, 62-66. In S. Haller and G. Simmons, eds. *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*. Menlo Park, CA: AAAI Press.

McCarthy, J. (chair) and Chaudri, V. (co-chair). 2004. *DARPA Workshop on Self Aware Computer Systems*. SRI Headquarters, Arlington, VA, Apr. 27-28.

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. diss. College of Computing, Georgia Institute of Technology.

Moorman, K., and Ram, A. 1995. Creativity in reading: Understanding novel concepts. In A. Ram and K. Moorman eds. *Understanding Language Understanding: Computational models of reading*, 359-395. Cambridge, MA: MIT Press/Bradford Books.

Murdock, J. W. 2001. *Self-improvement through self-understanding: Model-based reflection for agent adaptation*. Ph.D. diss., College of Computing, Georgia Institute of Technology.

Murdock, J. W. and Goel, A. K. 2001. Meta-Case-Based Reasoning: Using Functional Models to Adapt Case-based Agents. In D. W. Aha, I. Watson, and Q. Yang. eds. *Case-Based Reasoning Research and Development: Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR-2001,* 407-421. Berlin: Springer.

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., 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 L. Reder ed., *Implicit memory and metacognition*, 45-77. Mahwah, NJ: LEA.

Rosenthal, D. 2000. Introspection and Self-Interpretation, *Philosophical Topics*, a special issue on introspection, *28*(2): 201-233.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach* (2ed). NJ: Prentice Hall.

Schank, R. C. 1979. Interestingness: Controlling Inferences. *Artificial Intelligence 12*, 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.

Veloso, M. M. 1994. *Planning and Learning by Analogical Reasoning*. Berlin: Springer.

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.