Metareasoning, Monitoring, and Self-Explanation

Michael T. Cox

BBN Technologies
Intelligent Computing
Cambridge, MA 02138
mcox@bbn.com

Abstract. This paper seeks to extend notions of monitoring in metareasoning to include symbolic and linguistic expressions of self for purposes of communication and learning. The essay is intended to present a synthesis in plain language that challenges the agent community interested in metareasoning to consider what it means for a system to understand itself in any meaningful way. The basic claim is that if an agent truly knows what it is doing and why, it should be able explain itself to others using natural language or some other interactive mechanism with humans. To perform self-explanation it must be able to understand itself, and for this to occur it must monitor its own metareasoning and have an episodic memory that forms the basis of self. A further challenge is to incorporate self-explanation into an evaluation function that complements criteria based solely on action performance.

1 Introduction

A significant research history exists with respect to metareasoning in agent-based systems [1,9], and much of it is driven by the problems of limited rationality. That is because of the size of the problem space, the limitations on resources, and the amount of uncertainty in the environment, only approximate solutions can be obtained for finite agents. So for example with an anytime algorithm that incrementally refines plans, an agent must choose between executing the current plan or further deliberation with the hope of improving the plan. To make this choice, the agent is reasoning about its own reasoning (i.e., planning) as well as its potential actions in the world (i.e., the plan). As such this represents the problem of explicit control of reasoning.

Figure 1 illustrates the control side of reasoning along its upper portion. Reasoning controls action at the ground level in the environment; whereas metareasoning controls the reasoning at the object level. For the anytime controller, metareasoning decides when reasoning is sufficient and thus action can proceed. Although other themes exist within the metareasoning tradition, this characterization is a common one (e.g., [20,36,42]).

The complementary side of metareasoning, however, is less well studied. The introspective monitoring of reasoning performance requires an agent to maintain some kind of internal feedback in addition to perception, so that it can perform effectively and can

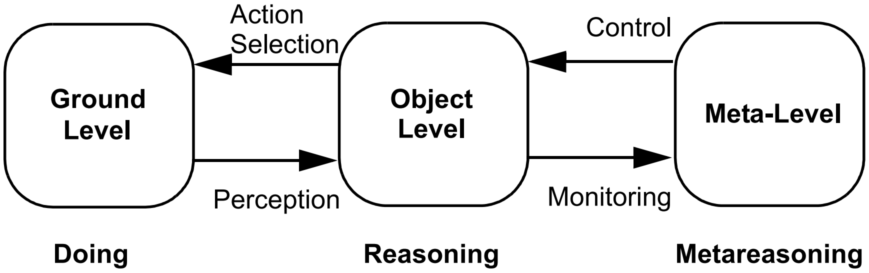

Figure 1 Duality in reasoning and acting

evaluate the results of metareasoning. For instance Zilberstein [53,54] maintains statistical profiles of past metareasoning choices and the associated performance and uses them to mediate the subsequent control and dynamic composition of reasoning processes.

But introspective monitoring can be even more explicit. If the reasoning that is performed at the object level and not just its results is represented in a declarative knowledge structure that captures the mental states and decision-making sequence, then these knowledge structures can themselves be passed to the meta-level for monitoring. For example the Meta-AQUA system [13] keeps a trace of its story understanding decisions in structures called a Trace Meta-eXplanation Pattern (TMXP). Here the object-level story understanding task is to explain anomalous or unusual events in a ground-level story perceived by the system.[1] Then if this explanation process fails, Meta-AQUA passes the TMXP and the current story representation to a learning subsystem. The learner performs an introspection of the trace to obtain an explanation of the explanation failure called an Introspective Meta-eXplanation Pattern (IMXP). The IMXPs are used to generate a set of learning goals that are passed back to control the object-level learning and hence improve subsequent understanding. TMXPs explain *how* reasoning occurs; IMXPs explain *why* reasoning fails.

Unfortunately these meta-explanation structures are so complicated that, although they have been shown empirically to support complex learning, they cannot be easily understood by humans. Indeed before I demonstrate the Meta-AQUA system to others, I often spend twenty minutes reviewing the TMXP and IMXP schemas, so that I can answer questions effectively. However I claim that all metareasoning systems share this characteristic. The kinds of recursive processing an agent must do to perform metareasoning (e.g., within the metacognitive loop of [2]) and the types of knowledge structures used to support metareasoning (e.g., the introspective explanations in [35] or [17]) produce a severe cognitive demand on even the most sophisticated observer. What is required is the implementation of an infrastructure to support interactive explanation of an agent's own reasoning.[2] By so building such an infrastructure, we not only improve our understanding of the design of intelligent agents, but we also move toward agents

[1] Note that no action-selection occurs with the story understanding task.

that truly understand what they are doing and why. A solution lies along the monitoring side of metareasoning.

This paper will examine further the potential that the monitoring of reasoning provides and will consider what implications exist. For metareasoning in agent-based systems, the self is the object of the processing, yet for many researchers the centrality of this statement is left wholly implicit. Here we will briefly discuss four characteristics or aspects of self from a computational stance. We consider in turn self-modifying code, self-knowledge, self-understanding, and finally self-explanation. After discussing the issue of evaluation, we will conclude by enumerating some outstanding problems related to metareasoning.

2 Self-Modifying Code

Like many novice programmers, I was fascinated by the idea of self-modifying code as an undergraduate. It seemed to capture in a direct and elegant way the idea of learning and intelligence. Of course my instructor quickly pointed out that this was a bad idea from a software engineering perspective and constituted a poor design. It generates hard to understand code that is very difficult to debug, because the flow of control lacks transparency. Instead the goal of top-down design is to abstract the environment using relevant data structures and to model the dynamics and interactions with these data structures. To effect a change in the behavior of the program, it was preferable to modify the data, not the code. Yet for some of us, it is easy to confuse techniques of self-modification with the principles of metareasoning.

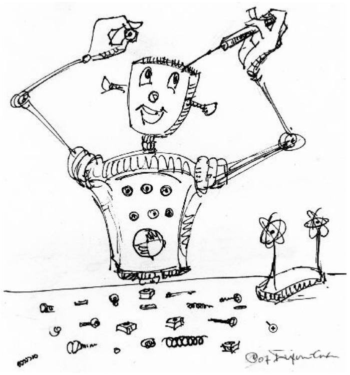

Figure 2 Self-modifying code

For example in the concluding paragraph of a chapter from an expert-systems textbook, Lenat, Davis, Doyle, Genesereth, Goldstein and Schrobe [28] proclaim the following:

[2] McGuinness and associates [32,33] have made a similar claim with respect to explanation for the semantic web.

> "Once self-description is a reality, the next logical step is self-modification. Small, self-modifying, automatic programming systems have existed for a decade; some large programs that modify themselves in very small ways also exist; and the first large fully self-describing and self-modifying programs are being built just now. The capability of machines has finally exceeded human cognitive capabilities in this dimension; it is now worth supplying and using meta-knowledge in large expert systems." (p. 238)

Given that this quote is nearly a quarter of a century old, we can attribute the over-enthusiastic response to some amount of naïveté, but this assertion remains astonishing nonetheless. To what dimension are they referring when they claim that human capabilities have been supplanted by machines and in what manner is self-modification and meta-knowledge the prime factors? Surely by most accounts, this exaggerates. The primary issue should be the relationship between learning and metareasoning.

To improve performance an agent must be able to adapt and change so that over time better decisions accrue. Changes can occur in essentially two ways. Agents are commonly construed as functions from a current state to some action that will change the state. Self-modification can be cast as adaptive changes to this function given as suitable representation for the function in a particular data structure and rigorous algorithms that transform the function. Alternatively learning can be cast as an accumulation of knowledge. As an agent acquires more knowledge and as its knowledge base is refined and reorganized, its performance and action selections should improve as a result. Yet it is unclear how metaknowledge is related to successful change and whether the two alternatives just described can be related.

3 Self-Knowledge

Many researchers stress the importance of metaknowledge in the design of intelligent agents and certainly many papers on metareasoning discuss metaknowledge (e.g., [3,4,9,15,18,35]). Metaknowledge being knowledge about knowledge seems at first blush to be crucial to learning if not action. That is how can an agent improve its knowledge without understanding the knowledge. Indeed the area of knowledge refinement appears to need much in addition to lone assertions in order to evaluate a knowledge base and to be able to make the necessary changes. Yet much of the recent trend in learning research demonstrates just how much an agent can learn using data-driven statistical approaches such as reinforcement learning.

Much ambiguity also exists with respect to metaknowledge and planning agents. Confusion results when a cognitive process such as planning and when knowledge concerning the world such as plans are mistaken for metacognitive processes and self-knowledge at the meta-level. Part of this problem is the fact that metaknowledge can exist at both the object and meta-levels and that interactions occur between levels. Consider the statement "The robosoccer agent followed its plan and won the championship because the plan was a good one." I claim that, although this statement contains metaknowledge, it does not necessarily involve metareasoning. Instead it refers to action at the ground level (i.e., soccer actions) controlled by an object-level constructed piece of

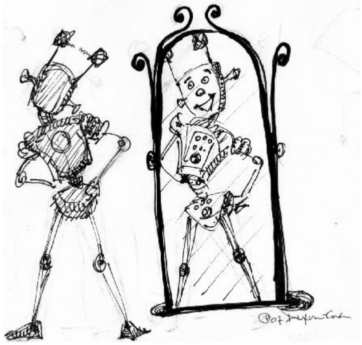

Figure 3 Self-knowledge

knowledge (i.e., the game plan). To state that the plan was a good one is an assertion about the plan and thus knowledge about knowledge, but at no point must we infer metareasoning or the meta-level. Thus metaknowledge is independent of metareasoning.[3]

Furthermore the statement concerns another agent and does not involve the self. Self-knowledge arises in part from the psychological distinction between semantic and episodic memory [50]. Semantic knowledge is general knowledge about objects such as "All psychologists know a lot about human thinking." Episodic knowledge concerns actual events or episodes in a person's life or in an agent's action history. Much of human reasoning is driven by this type of concrete experience. For example I might know that all computer scientists are good at mathematics and that I am a computer scientist. But I would not conclude that I am good at math through logical deduction with this semantic knowledge. Instead I have many experiences with performing mathematics and have come to trust my ability to do similar problems in the future. Such confidence in my own ability is metaknowledge derived by reasoning about my own reasoning experiences.

Such an approach to self-knowledge uses a case-based reasoning [23][26][30][39] perspective. That is a case-based agent performs reasoning by being reminded of past cases of experience and adapting these cases to the current situation when interpreting perceptions (case-based understanding) or choosing an action to perform (case-based planning). Ironically and like most AI programs, few case-based implementations focus on an explicit representation of the self or otherwise operationalize the self despite specific case libraries that represent experience.[4]

[3] Note that this statement is an assertion about metaknowledge and therefore meta-meta-knowledge. However this distinction is not necessarily that important or useful. What is important is simply that we as researchers be clear with our categories for purposes of *communication* of agent design and implementation.

[4] An interesting exception exists with the research of Forbus and Hinrichs [16] that track agent activity logs to ascertain episodic information with respect to self-knowledge.

4 Self-Understanding

Early research in the case-based reasoning community concentrated upon cognitive modeling of the human comprehension process, especially in terms of how humans acquire conceptual understanding of stories or textual representations (e.g., [44,45]). As is the case with Meta-AQUA, the story understanding task is to take as input a representation (either conceptual or textual) of the story and to output an interpretation of the input. Although interpretation can take many forms, the CBR stance is to retrieve a piece of experience (i.e., a script or case) that matches the content of the current sentence and to adapt it to produce the interpretive understanding. So the story is understood, if the program can successfully answer questions about the story, paraphrase it, or connect the representations into a coherent whole that predicts further events in the story. More generally this same process can be applied to monitor one's own plans or exogenous events executed in the world or to monitor reasoning performed in the head. The key is that monitoring like control is a first class citizen in both the reasoning and metareasoning processes [11,48].

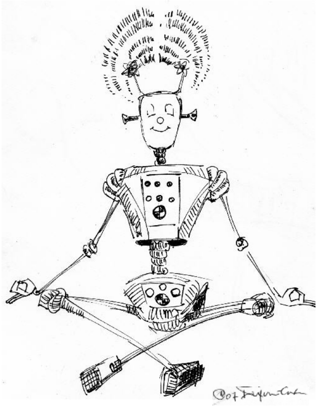

Figure 4 Self-understanding

An INitial inTROspective cognitive agent called INTRO [8] combines planning and understanding within a Wumpus World environment by integrating the PRODIGY planning and learning architecture [7,52] with the Meta-AQUA story understanding and learning system. Rather than input all goals for the agent to achieve, the understanding component compares expected states and events in the world with those actually perceived to create an interpretation. When the interpretation discloses divergence from those expectations, INTRO generates its own goals to resolve the conflict. These new goals are then passed back to the PRODIGY component so that a plan can be generated and then executed. As such introspective monitoring controls action through the creation of new goals.

This understanding process depends upon declaratively represented percepts of ground-level states and actions. If the reasoning processes at the object level (i.e., the mental states and inferences) are likewise represented declaratively, metareasoning can monitor such activity to obtain some measure of self-understanding. As mentioned in the introduction, the Meta-AQUA system implements a theory of introspective multistrategy learning whereby the system builds and executes a learning plan to achieve a set of learning goals. These goals are spawned in response to explanations of explanation failures which allows the system to decide what to learn. However much remains to be implemented in the INTRO system to achieve a full integration of reasoning and metareasoning and of world knowledge and self-knowledge.

For example other than with goal generation, monitoring has no other control over INTRO's reasoning. Consider the possible responses to a failed robosoccer plan. If an agent was to reason about why its game plan did not succeed by considering its prior planning (e.g., "I focused on our ball-handling when creating the plan rather than the defender's capabilities.") as opposed to simply analysing the plan or the plan execution, then metareasoning and monitoring would be involved. But INTRO cannot use such inference to improve its future planning performance. Furthermore planning itself is not influenced by cases of prior planning, although an introspective version of PRODIGY called Prodigy/Analogy [51] has that capability. Certainly INTRO has never felt a familiarity at planning time so that it might say to itself "This partial plan must be close to a correct one, because I have performed similar planning before."[5] Also INTRO cannot decide whether it is competent enough for a task (at either the ground or object level) or whether it should ask another agent to perform the task instead. Finally despite the fact that INTRO might invoke the Meta-AQUA component to explain some failed reasoning, it cannot actually explain the failure to you. Here I claim that not only INTRO but all metareasoning agents would benefit from similar capabilities.

5 Self-Explanation

Explanation-aware computing is seeing a recent resurgence in the AI and cognitive science communities as indicated by the existence of main-stream workshops [40,41] and compilations [22]. Explanations provide numerous functions including event prediction, assignment of personal (e.g., legal) blame, and diagnosis for repair [21], but their most central purpose is to determine causal connectedness in service of learning [21,38,43]. Explanation provides a key capability for elaborating the understanding that agents produce when processing the environment, especially when agents' perceptions diverge from their expectations. An explanation likewise provides a causal accounting of mental anomalies discovered during monitoring.

Explanation is ubiquitous. While discussing self-knowledge, I provided an explanatory sentence as an example. To assert that a robosoccer agent won a championship because the plan was a good one is to causally link the characteristics of the plan to successful performance (i.e., following the plan). A story understanding agent comprehends an input, if it can explain why the characters in the story do surprising things by inferring what their goals and motivations are and by enumerating those events that follow from earlier ones in a causally determined manner (e.g., event e1 results in a state that is the precondition or determinant for another event e2). As such explanations link

[5] But see the Dial case-based planner [27] that considers familiarity with past reasoning.

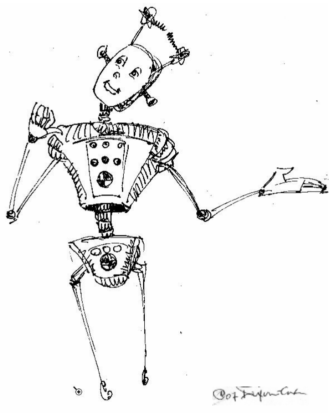

Figure 5 Self-explanation

the causally relevant past events with the desired future states to account for current observations.[6] Yet explanation is all too absent in many agent-based implementations.

5.1 Like rats in a maze

As commented upon in section 3, statistical techniques have proven to be a powerful technique that enables agent-based systems to learn complex behaviors from interactions with regularities in the environment. Indeed given the Markov assumption of independent decisions, we can model an agent with the policy π *(s) that returns an optimal action, a, from any given state, s. Even though an agent may not know the environmental probability distribution between states and actions, an agent that explores states through its actions can converge on the optimal policy by experiencing rewards and using Q-learning.

The technique has been used in many complex situations and under various conditions of uncertainty to model behavior in the natural world. For example both Konidaris and Hayes [24] and Sharma [47] have shown reinforcement learning to be capable of simulating the maze learning behavior of rats. In the simplest of trials, one can place a rat at the base of a T-maze with cheese in one of the two arms. The task is for the rat to find the food. Given a sufficient number of trials, the algorithms will learn the correct set of actions to find the reward. More complex mazes can be assembled by connecting multiple Ts, each representing a binary decision. But consider the following. The artificial rats may learn to find the reward, but do they know where the cheese is?

Many years ago Tolman [49] provided a very interesting experiment at Berkeley with actual rats. They had two groups that experienced very different conditions. The first group of rats represented the standard condition. These rats were deprived a food for a length of time so that they were hungry, and they were trained over an eleven day period through a complex maze system. By the end of the period they had reached a

[6] Such *self-projection* into the future and past has been linked to central human cognitive abilities [6].

high level of performance so that they made few errors when running the maze. The second group of rats were fed until satiation and then were strapped into small wheelbarrows. The experimenters then pushed the wheelbarrows through the maze to the location of the cheese. After eleven days the second group was tested using a standard test (hungry and on foot). The surprising discovery was that the group very quickly gained performance equal to that of the standard learning condition. The experiment demonstrated latent learning in the absence of reward. That is the reward was necessary for performance but not for learning!

This is relevant to metareasoning, because many forms of metareasoning use data-driven statistical methods and reinforcement learning driven only by performance as determined by a reward schedule (e.g., [19,36]). Moreover it is difficult to claim that these systems can understand themselves in an explicit way, although they have a statistical model of their own reasoning and reason recursively about the model. This is true at both the object and at the ground levels. The reinforcement-learning agent reported by Anderson and colleagues [2,46] consists of an internal metacognitive loop that detects when the rewards in the environment diverge from its expectations. Analysis of such perturbations lead to improved performance with respect to standard reinforcement learners. But can this type of agent explain how and why it learns? Because the statistical models have no symbolic content, explanation is handicapped. Instead we should consider how an agent might learn an explainable policy $\pi^e(s)$ that decides to take action, a, when in state, s, because justification, j. For example such a policy would suggest that the rat turn left at the T junction, *because the cheese is at the end of the left arm*. When the cheese is no longer to be found to the left and is instead at the end of the right arm, a straightforward explanation of failure should result in more effective learning. Granted Raja and Goel [35] are making progress toward enabling introspective explanations, but as mentioned previously, the kinds of explanations structures used in metareasoning (i.e., meta-explanations and introspective explanations) are of less use to humans trying to understand the metareasoning.

5.2 From rats to cognitive agents

Two characteristics separate humans from all other species including rats. First is our creative use of natural language and our ability to communicate to others (and to ourselves). Second is the (albeit limited) ability to introspect and to explain our identity as an individual. This paper challenges the metareasoning community to develop computational frameworks within which these two characteristics synthesize. The goal is to create cognitive agents that can explain themselves to others in plain English. Evidence exists that such self-explanation behavior can help agents learn better [8,13] and lucid English translations will clearly help humans gain trust in their cognitive assistants.

The general problem faced by users of cognitive systems is that of trust calibration. Some users overestimate the ability of systems whereas others underestimate or mistrust them. The root cause in both cases is that users do not understand how or why computers do what they do. If a system could explain itself in English and tell a user why it makes a particular decision, the user will more likely know the correct uses and limits of that system. But most importantly, it is the very act of explaining itself that allows a system to improve its performance in ways that ordinary machine learning programs never will.

The problem faced by designers of cognitive systems is that the intelligent agents they wish to develop are so complicated that existing and foreseeable design techniques are unable to effectively engineer them with existing technology alone. Yet if a learning agent could participate in its own testing and debugging, the agent might explain those components of its software that have implementation failures so that engineering bottlenecks can be overcome. One direction toward this ideal is to formulate systems that generate detailed explanation graph structures of their internal behavior and provide interactive graph navigation aids with English generation abilities.

Many reasons exist for self-explanation, but it is not an easy task. Table 1 lists my top ten favorite explanations I would like to hear a cognitive agent communicate. Consider the first. Many humans have explained to their friends that they were late for an appointment, because they forgot to fillup their car with gas. Cox [12] notes that, if a case-based planner uses an indexed memory for retrieval of past cases in lieu of exhaustive search, then forgetting is a potential causal factor in planning failures.

Table 1: Ten simple mental explanations

1. I forgot that X.
2. I am good at Y.
3. I did not see (or notice) Z.
4. I mistook an M for an N.
5. I assumed that I is the case because B.
6. I thought that all J could K.
7. I learned that Q today.
8. I did not have enough time to think about R.
   I wasted time worrying (thinking) about R.
9. S surprised me because B.
10. I chose to do A1 instead of A2 because B.
    I wanted to achieve G1 rather than G2 because B.

Figure 6 illustrates the IMXP explanation structure for such a reasoning failure. The language task then is to take this graph as input and to output either a paraphrase or elaboration in English text. A suitable paraphrase might be “I forgot to fill up the car with gas when I was at the store.” An elaboration might be something similar to the following text. “The context, C, of being at the store did not sufficiently match the index, I, with which the goal, G, to fill up with gas was stored in memory, so I failed to retrieve the goal at the right time and thus did not put gas in the tank. Because the tank was low, I did not have enough fuel and then ran out of gas.” Being able to generate such text might be possible using existing language generation algorithms (e.g., [29,31]), although many problems of generative focus exist.

Another open research question remains as to the best method of quantitatively evaluating subjective explanation. An explanation can be true but totally miss the point.

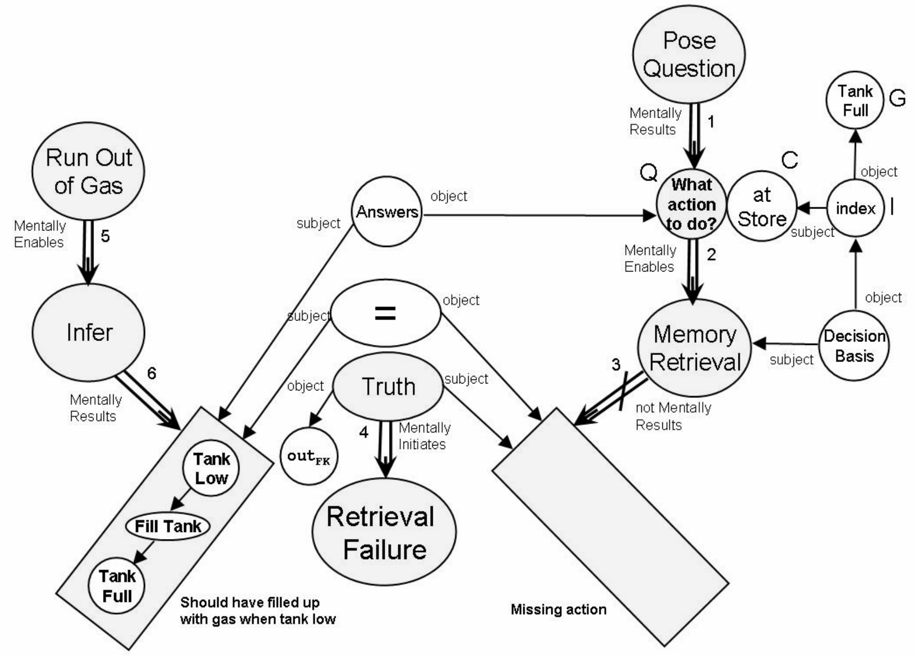

Figure 6 "Forgetting to fillup with Gas" Meta-XP structure

For example it does not help to understand why firemen wear red suspenders by explaining that the suspenders hold their pants up [37]. What is most important in evaluating an explanation is not its veracity per se, but whether it serves the need of the agent (either self or other) targeted by the explanation. The need is in terms of the current knowledge of the agent and gaps in the knowledge that the explanation fills [25,38]. So when knowledge is missing, incorrect, or disconnected from related knowledge, the best explanation fills the gap, corrects the misconception, or causally links assertions that provide further coherence and relational structure.

The challenge is to take these long-standing, subjective principles and to operationalize them constructively. A numeric criterion may be a poor substitute for evaluating the extent to which a large graph structure fulfils conceptual needs as opposed to strictly syntactic ones (e.g., the number of connected components in an explanation), but the loss due to abstraction and approximation is compensated by the ability to compare and contrast explanatory solutions. In a very simple way, the dissertation research of Cox [10,11] provides a start toward this goal. Each anomaly in a story represents a source of knowledge discrepancy for Meta-AQUA and a potential explanation target. For each anomaly up to three points are awarded: one point for identifying that a question needs to be posed, a second for providing any explanation, and a third for matching the "correct" explanation as enumerated by an oracle. With this or any like function, the evaluator should generate a real number between 1 and 0. Then to normalize the explanation criterion with performance (given a performance measure also between 1 and 0), it is sufficient to calculate performance/(2-explanation). When explanation is 1, the measure reflects performance alone; otherwise the measure can be reduced by as much as half the normal performance. Without the incorporation of self-explanation into the overall performance measure, many metareasoning implementations can simply optimize performance first and then sprinkle on a bit of meta-sugar after the fact.

6 Conclusion

I am not the first to call for agents that truly know what they are doing and why. Raja and Goel [35] make many of the same arguments, and Brachman [5] issued the challenge beforehand. Indeed Brachman initiated DARPA's Cognitive Computing vision that seeks to solve basic research and development problems related to those described here [14]. One difference is that I claim that, if an agent really understands what it is doing and why, then it should be able to explain this self-understanding to others as well.

In conclusion I will simply close with a list of hard problems that, in addition to the problems of text generation and evaluation, seriously impede progress toward agents that can meaningfully claim to know themselves.

1. *The Problem of Appropriateness*: Given that metareasoning creates an additional computational burden, how can an agent decide when the potential benefit of metareasoning will outweigh the cost of its overhead?
2. *The Homunculous Problem*: How can we effectively control metareasoning without substituting yet another computational layer above the meta-level?
3. *The Problem of Consciousness*: How can the many heterogeneous reasoning functions such as problem solving, understanding, learning be multiplexed with metareasoning into a whole that represents the unity of experience?
4. *The Existential Problem*: What are the computational properties that lie beneath the illusion of separate, independent existence and free will?[7]
5. *The Problem of Identity*: What knowledge structure best represents the abstract notion of self?

Acknowledgements

I thank David Leake, Shlomo Zilberstein and the anonymous reviewers for criticism of an earlier draft. Deborah Dixon did much to lighten up this essay with her illustrations.

References

[1] Anderson, M., & Oates, T. (2007). A review of recent research in metareasoning and metalearning. *AI Magazine 28*(1): 7-16.

[2] Anderson, M., & Perlis, D. (2005). Logic, self-awareness and self-improvement: The metacognitive loop and the problem of brittleness. *Journal of Logic and Computation*, 15(1): 21-40.

[7] Other than in Minsky (1965), I know of no other comment on the computational aspects of free will.

[3] Arkoudas, K., & Bringsjord, S. (2005). Metareasoning for Multi-agent Epistemic Logics. In J. Leite & P. Torroni (Eds.), *Computational Logic in Multi-Agent Systems: 5th International Workshop, CLIMA V, Lisbon, Portugal, September 29-30, 2004* (pp. 111-125). Berlin: Springer.

[4] Barklund, J., Dell'Acqua, P., Constantini, S. & Lanzarone, G. A.(2000) Reflection principles in computational logic. *Journal of Logic and Computation, 10*(6): 743-786.

[5] Brachman, R. J. (2002, Nov/Dec). Systems that know what they are doing. *IEEE Intelligent Systems* 67-71.

[6] Buckner, R. L., & Carroll, D. C. (2007). Self-projection and the brain. *Trends in Cognitive Sciences 11*(2): 49-57.

[7] Carbonell, J. G.; Knoblock, C. A.; and Minton, S. (1991). PRODIGY: An integrated architecture for planning and learning. In K. VanLehn ed., *Architectures for intelligence: The 22nd Carnegie Mellon symposium on cognition*, 241-278. Hillsdale, NJ: Lawrence Erlbaum Associates.

[8] Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI Magazine 28*(1): 32-45.

[9] Cox, M. T. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence 169*(2), 104-141.

[10] Cox, M. T. (1996a). An empirical study of computational introspection: Evaluating introspective multistrategy learning in the Meta-AQUA system. In R. S. Michalski & J. Wnek, (Eds.), *Proceedings of the Third International Workshop on Multistrategy Learning* (pp. 135-146). Menlo Park, CA: AAAI Press. (Available at `http://mcox.org/Ftp/eval-paper.ps.Z`)

[11] Cox, M. T. (1996b). *Introspective multistrategy learning: Constructing a learning strategy under reasoning failure*. (Tech. Rep. No. GIT-CC-96-06). Doctoral dissertation, Georgia Institute of Technology, College of Computing, Atlanta. (Available at `http://hcs.bbn.com/personnel/Cox/thesis/`)

[12] Cox, M. T. (1994). Machines that forget: Learning from retrieval failure of mis-indexed explanations. In A. Ram and K. Eiselt eds. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society,* 225-230. Hillsdale, NJ: LEA. (Available at `http://mcox.org/Papers/mach-forget.ps.gz`)

[13] Cox, M. T., & Ram, A. (1999). Introspective multistrategy learning: On the construction of learning strategies. *Artificial Intelligence, 112*, 1-55.

[14] Defence Advanced Research Projects Agency (2007). *Strategic plan*. US Government.

[15] Davis, R. (1980). Meta-rules: Reasoning about control. *Artificial Intelligence 15*: 179-222.

[16] Forbus, K., & Hinrichs, T. (2004). Companion cognitive systems: A step towards human-level AI. In *AAAI Fall Symposium on Achieving Human-level Intelligence through Integrated Systems and Research*, October, Washington, DC.

[17] Fox, S., & Leake, D. (2001). Introspective reasoning for index refinement in case-based reasoning. *Journal of Experimental and Theoretical Artificial Intelligence 13*(1):63-88.

[18] Hahn, U., Klenner, M., & Schnattinger, K. (1996) Automated knowledge acquisition meets metareasoning: Incremental quality assessment of concept hypotheses during text understanding (pp. 9--14). In *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*.

[19] Hansen, E. A., & Zilberstein, S. (2001). Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence 126*(1-2): 139-157.

[20] Horvitz, E. (1987). Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of the Third Workshop on Uncertainty in Artificial Intelligence* (pp. 429-444). Seattle, Washington.

[21] Keil, F. C. (2006). Explanation and understanding. *Annual Review of Psychology 57*: 227-254.

[22] Keil, F. C., & Wilson, R. A. (Eds.) (2000). *Explanation and cognition*. Cambridge, MA: MIT Press.

[23] Kolodner, J. L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

[24] Konidaris, G. D., & Hayes, G. M. (2005). An architecture for behavior-based reinforcement learning. *Adaptive Behavior 13*(5).

[25] Leake, D. (1994). Accepter: Evaluating explanations. In R. C. Schank, A. Kass, & C. K. Riesbeck (Eds.), *Inside case-based explanation* (pp. 168-206). Hillsdale, NJ: Lawrence Erlbaum Associates.

[26] Leake, D. (Ed.) (1996). *Case-based reasoning: Experiences, lessons, and future directions*. Menlo Park: AAAI Press/MIT Press.

[27] Leake, D., Kinley, A., & Wilson, D. (1996). Linking adaptation and similarity learning (pp. 591-596). In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: LEA.

[28] Lenat, D. B., Davis, R., Doyle, J., Genesereth, M., Goldstein, I., & Schrobe, H. (1983). Reasoning about reasoning. In F. Hayes-Roth, D. A. Waterman, & D. B. Lenat (Eds.), *Building expert systems* (pp. 219-239). London: Addison-Wesley Publishing.

[29] Lester, J., & Porter, B. (1996). Scaling up explanation generation: Large-scale knowledge bases and empirical studies (pp. 416-423). In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. Menlo Park: AAAI Press/MIT Press.

[30] Lopez de Mántaras, R., 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.

[31] McDonald, D. (1993). Issues in the choice of a source for natural language generation. *Computational Linguistics 19*(1): 191-197.

[32] McGuinness, D. L., Ding, L., Glass, A., Chang, C., Zeng, H., & Furtado, V. (2006). Explanation interfaces for the semantic web: Issues and models. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop* (collocated with ISWC 2006).

[33] McGuinness, D., & Patel-Schneider, P. (2003). Infrastructure for web explanations. In *Proceedings of 2nd International Semantic Web Conference (ISWC2003*).

[34] Minsky, M. L. (1965). Matter, mind, and models. In *Proceedings of the International Federation of Information Processing Congress 1965* (Vol. 1) (pp. 45-49).

[35] Raja, A., & Goel, A. (2007). Introspective self-explanation in analytical agents. *This volume*.

[36] Raja, A., & Lesser, V. (2007). A framework for meta-level control in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*. (Available at `http://www.springerlink.com/content/3721815268763445`)

[37] Ram, A. (1989). *Question-driven understanding: An integrated theory of story understanding, memory and learning* (Tech. Rep. No. 710). Doctoral dissertation, Yale University, Department of Computer Science, New Haven, CT.

[38] Ram, A., & Leake, D. (1991). Evaluation of explanatory hypotheses. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp. 867-871). Hillsdale, NJ: Lawrence Erlbaum Associates.

[39] Riesbeck, C. K., & Schank, R. C. (Eds.) (1989). *Inside case-based reasoning.* Hillsdale, NJ: Lawrence Erlbaum Associates.

[40] Roth-Berghofer, T. R., (Ed.) (2005). *Explanation-Aware Computing: Papers from the AAAI Fall Symposium*. Technical Report FS-05-04. Menlo Park, CA: AAAI Press.

[41] Roth-Berghofer, T. R., Schultz, S., & Leake, D. B. (Eds.) (in press). *Proceedings of the AAAI-07 Workshop on Explanation-aware Computing.* Menlo Park, CA: AAAI Press.

[42] Russell, S. J., & Wefald, E. (1991). Principles of metareasoning. *Artificial Intelligence 49:* 361-395.

[43] Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[44] Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[45] Schank, R. C., & Riesbeck, C. (Eds.). (1981). *Inside computer understanding: Five programs plus miniatures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[46] Schmill, M., Josyula, D., Anderson, M. L., Wilson, S., Oates, T., Perlis, D., & Fults, S. (2007). Ontologies for reasoning about failures in AI systems. *This volume*.

[47] Sharma, R. (2003). Latent learning in agents. In *Proceedings of First Instructional Conference on Machine Learning*.

[48] So, R., & Sonenberg, L. (2007). Situation awareness as a form of meta-level control. *This volume.*

[49] Tolman, E. C. (1948). Cognitive maps in rats and man. *Psychological Review 55*: 189–208.

[50] Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory*, (pp. 381-403). New York: Academic Press.

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

[52] Veloso, M., Carbonell, J. G., Perez, A., Borrajo, D., Fink, E., & Blythe, J. (1995). Integrating planning and learning: The PRODIGY architecture. *Journal of Theoretical and Experimental Artificial Intelligence*, *7*(1): 81-120.

[53] Zilberstein, S. (1993). *Operational rationality through compilation of anytime algorithms*. Ph.D. Dissertation, University of California at Berkeley.

[54] Zilberstein, S., & Russell, S. J. (1996). Optimal composition of real-time systems. *Artificial Intelligence 82*(1-2):181-213.