To BICA and Beyond: RAH-RAH-RAH!

—or—

How Biology and Anomalies Together Contribute to Flexible Cognition

Don Perlis

Department of Computer Science, University of Maryland
College Park, MD 20742, USA
perlis@cs.umd.edu

Abstract

Evolutionary and cognitive examples are used to motivate an approach to the brittleness problem and automated flexible cognition, centering on the notion of an anomaly as the key focus of processing.

Introduction

Somehow, living organisms deal with a complex world. This is the basis for the AI hope of smart systems worthy of that designation. But we have been unsuccessful at this so far. Here I will argue that a closer—but straightforward—look at human reasoning suggests an implementable strategy for flexible commonsense reasoning. The notion of an anomaly—something that deviates from expectations—is the central focus. I will being with motivating discussion, then examples, and a sketch of where further work is needed.

Anomalies are the problem

Why do cognitive architectures need inspiration? Simply put, because our efforts to design such architectures have, to date, been flops. Somehow we are missing a key ingredient; our most touted successes (think of Deep Blue) are idiot savants, brilliant at one thing and utterly clueless at everything else, even at minor variations on what they are good at; see (Anderson et al 2006b). But perhaps this is the silver lining: our artifactual systems lack the ability to deal with variation, perturbation, unexpected twists. In fact, this has a name: the brittleness problem. Brittle systems break when given a twist, when forced into circumstances beyond what they were explicitly designed for, i.e., when they encounter anomalies.

This means our systems are far less useful than we would like, than we in fact need in order to deploy them usefully in realistic settings. For the real world abounds in the unexpected. And dealing usefully with the unexpected is a hallmark of flexible human-level intelligence, whereas breaking in the face of the unexpected (wasting time, getting nowhere, brooking disaster) is the hallmark of current automated systems. Rational anomaly-handling (RAH) is then the missing ingredient, the missing link between all our fancy idiot-savant software and human-level performance.

To forestall a misunderstanding: RAH is not a matter of being clever or insightful. All humans have RAH (in varying degrees!); essentially it amounts to not being idiotic—not being blind to signs that things are getting a bit unusual.

But what is hidden in this seemingly elementary capability? For one, the unusual (i.e., the occurrence of an anomaly) is detected in virtue of its deviating from the usual, from the everyday expected set of conditions. Thus RAH involves having expectations, and having the means to compare them to observations as to what is happening. We will examine this a greater length in a subsequent section, but what has been said so far already sets the tone for what is to come.

Fumbling our way toward RAH

Why has this been so hard? After all, brittleness is not a newly discovered problem. Here are some parts of an explanation:

A. It has been very tempting, irresistibly so in many cases, simply to try to build in useful responses to individual anomalies. For instance, the enormous—and enormously impressive—literature on nonmonotonic reasoning appears to be an attempt to address the problem of providing a precise formal account of the complexities of the natural world, by setting out in advance what is normal and what is anomalous in a given context, and what is normal for an anomalous subcontext of a normal context, etc. As an example: birds (normally) fly; but penguins are abnormal (anomalous) within the flying-bird context, that is, penguins don't (normally) fly; but a propeller-outfitted one is abnormal with the non-flying-penguin subcontext; etc.

Such an approach appears to set out what to conclude about each and every object and context, no matter how unexpected. Yet this is doomed to failure, if on its own. There is no way to know in advance what the various anomalous subcontexts are—the world really is too surprising. Somehow the knowledge as to what the various subcontexts are has to be found from experience, and cannot be set out in advance. Once learned, then to be sure a very useful formal account can be given (perhaps revised later on, as conditions or information change). But this leaves wide open what to do when an as yet unlearned anomaly rears its head.

B. So-called adaptive systems—genetic algorithms, neural networks, etc—do learn, but very slowly, and—more to the point—do not make decisions about what, when, or how to learn (largely because they have no knowledge of why to learn in the first place—no recognition that there is an anomaly to be addressed). Still, the existence of such systems may have lulled us into thinking that learning was a well-explored area. It is and it is not: machine learning, including the aforementioned work on adaptive systems, is a major area of vigorous investigation and a very great deal of important results are now available. But the bulk of this work focuses on the learning per se, and not on the contexts within which learning is needed. Yet the latter is precisely what RAH requires.

C. Anomalies—that is, mismatches between expectations and observations—can be reasonably regarded as contradictions in a belief base: an agent has the belief *P* since *P* is expected, and also *-P* since the latter is "observed" (that is to say, *-P* is provided by some other source). Both cannot be true, both cannot be rationally believed, so the agent must do something about the situation. But traditional formal approaches to commonsense reasoning—including nonmonotonic logics—are hopelessly inappropriate in such cases, i.e., wben applied to inconsistent belief bases they "explode" to produce all wffs as theorems.

Thus efforts to capture human-level commonsense reasoning, with its marvelous RAH flexibility, have not been successful to date. It makes sense, then, to take a closer look at this human capability, in the hope that we might ferret out some key aspects that can be borrowed and automated. That is to say, perhaps artifactuality can recapitulate biology—the only known supplier of RAH.

Biology is the (only) answer (so far)

How has biology solved this problem? How do organisms cope with anomalies? What anomalies are organisms faced with?

Basically, an organism has processes that provide certain life-preserving benefits; and anything that rubs strongly against the grain of these processes is potentially something to be reckoned with, something that must be dealt with or the organism dies. And indeed that often is what happens, the survivors being those lucky few whose processes are less strongly rubbed against.

For instance, supposed "food" that is not healthful may result in survival of only those few members of a species that either safely metabolize that item or that avoid it. Thus anomalies drive evolution.

But this is not yet RAH; no rational process, no actual reckoning, is given here. Rather this is slow evolution, parameter adjusting, trial-and-error, as in adaptive computation. For a species, anomalies present a kind of challenge to rise to, but for an organism an anomaly is simply the end of the line, or a routine matter-of-course; there is no change, no learning, at the organism level—at least not from such "selective pressures".

Yet biology has not stopped with (Darwinian selective) evolution. For among the products of such evolution are mammals, primates, humans, creatures with RAH, agents that face anomalies and deal with them, that get better over time, and that do so in part by reasoning. What then is it that we do? And is it something that can be automated? Or does human-style RAH have no concise principles, instead being a mishmash product of evolution akin to a highly distributed neural network? In the latter case, we might manage to evolve our own RAH-capable algorithms but not know how they work, perhaps useful but not intellectually satisfying. Fortunately, there is evidence that it is the former, nor the latter, that obtains.

What principles underlie RAH, and how can they be transitioned from the biological to the artificial?

How does natural/biological RAH work? What is it that we humans do when we encounter and deal with an anomaly? The answer is surprisingly simple, even obvious—and also one that is borne out by work in cognitive psychology and neuroscience. It consists of five parts, which together define what we call the Metacognitive Loop (MCL):

(i) We have expectations as to how things will be.

(ii) We compare expectation to observation and thereby *note* indications that an expectation has been violated.

(iii) We *assess* what we know that might explain this violation.

(iv) We decide what response—if any—to *guide* into place.

(v) We revise expectations as needed.

It is introspectively clear that we do this—and do it a lot, every single day—and that without these five capabilities we would not do well at all, i.e., they are necessary parts of our job of being RAH agents. Whether the converse holds—that these five are sufficient for RAH—is perhaps less obvious. But at least this now puts us in a good position: (i)—(v) can be automated and studied, and the

sufficiency hypothesis—let's call it the MCL hypothesis—tested.

Below I will recount some such experiments; but first I need to clarify some of the five parts a bit. Let me start with item (iv), things that can be done about an anomaly. Recall my warning at the beginning: cleverness has little to do with RAH. So the things we can do generally are not ones requiring heavy duty analysis or insight, yet they must on balance be useful. These include such actions as asking for help, postponing, giving up, using trial and error, initiating training, double-checking our observations, seeking corroborating evidence, and so on. An implicit part of our hypothesis is that there is a relatively small "core" of anomaly-types and anomaly-resolutions, suited to virtually all domains. We currently are investigating ontologies of such, but work to date is highly suggestive that such a core exists. See (Schmill et al, 2007; Anderson et al, 2007).

Item (i) requires us to have ideas about how the world works, and this in turn requires some sort of learning capacity, as also does the "initiate training" option in (iv). In (ii) there is an implicit use for formal logics that behave "responsibly" in the presence of contradictions.

Thus (i)—(v) are not trivial, they come with substantial requirements. But they also lend themselves to algorithmic implementation. The training option is particularly interesting, for it dramatizes the need to decide that learning is needed, what is to be learned, and when, and how, and for how long. Thus an RAH-capable system will, among other things, need trainable modules (neural networks, reinforcement learners, etc). RAH then is not a substitute for traditional aspects of AI so much as an enhancement and bringing together of them.

A spate of RAH-capable systems

In the past several years, my group has been hard at work building and testing examples of RAH-capable systems. These include systems that perform (robot) navigation, reinforcement learning, nonmonotonic reasoning, video-arcade tank game playing, and human-computer natural-language dialog. Each of these was built separately, for that particular application. And each such system performed markedly better when its RAH aspects were employed. However, we soon realized that the same elements (i)—(v) were at work in some fashion or other in each case, and this led us to hypothesize a general-purpose domain-independent MCL module, akin to the human RAH that seems to work pretty well across the board, not just in two-player games, not just in preparing a meal, not just on the job. Any of us would probably do well even if spirited away—as an involuntary immigrant—to Finland in the dead of the night; we'd be astonished and deeply worried, but we'd find a way to survive, make our concerns known, even in another language, and we'd eventually manage to get back home—or we'd become adapted to Finnish life. So why should an automated RAH be any less domain-independent? We now take a closer look at some of our MCL-enhanced systems (in particular our natural-language system), at a more ambitious project we are poised to launch, and at still broader requirements for fully human-level RAH.

Reinforcement learning (RL) is a well-established methodology that works very well in many settings, notably ones in which the reward structure is static or nearly static. But when that structure is changed suddenly and significantly, the performance of RL degrades severely and recovers excruciatingly slowly. In essence, RL algorithms need to ``unlearn'' what they have learned, step by step, since they have no way to recognize that the reward structure has changed, let alone assess what can be done about it. Yet it is clear that, given a drastic change that makes previous learning useless, the best policy is simply to throw it out and start over.

Using a variety of reinforcement learning algorithms (Q-learning, SARSA, and prioritized sweeping) we experimented with a simple 8x8 grid world with rewards in cells (1, 1) and (8, 8). The learner was trained for 10,000 steps, then the rewards were switched and learning continued for another 10,000 steps. We compared the performance of standard RL algorithms to MCL-enhanced versions of the same algorithms. The MCL-enhanced RL algorithms maintained and monitored expectations about such things as average reward per step, value of future rewards, and average time to next reward. When these expectations were violated, they assessed the nature of the violation and, using a simple decision tree, chose one of the available repairs. These included: ignoring the problem, adjusting the learning parameter, or throwing out the current action policy and starting over.

Performance rises sharply and levels off until step 10,000 when the reward-switching occurs. At that point, performance falls dramatically and then begins to recover. However, the standard RL algorithms recover far more slowly and far less completely than the MCL-enhanced versions (the higher curve) of the same algorithms. In our experiments we found that the greater the degree of change in reward (such as swapping rewards for penalties, and vice versa), the greater the benefits generated by MCL. See (Anderson et al, 2006a).

Bolo is a multi-player tank game which takes place in a world that contains various terrain types (roads, swamps, walls, etc.), refueling bases, and pillboxes. There are three types of pillbox: neutral pillboxes fire on all tanks, dead pillboxes pose no threat and can be captured to make them friendly, and friendly pillboxes fire only on other players' tanks. An important strategy in Bolo is to capture pillboxes, make them friendly, and then use them either offensively or defensively.

Bolo can be played by humans, but it can also be played by programs. Such artificial Bolo players tend to play quite poorly and are easily fooled when unexpected complications arise (change of terrain, more dangerous pillboxes, etc). Thus Bolo provides a good challenge domain in which to test MCL.

Our MCL-enhanced Bolo player is controlled by a simple Hierarchical Task Network (HTN) planner with primitive actions that ground out in controllers. It maintains a variety of expectations, the primary one being that the tank it controls will not be destroyed. Over time it learns from its mistakes, first discovering that its performance is poor, and then using a form of trial-and error to find a way to improve. See (Schmill et al, 2008).

Natural language—and especially natural language human-computer dialog—is arguably the most difficult application we have explored to date. Natural language is complex and ambiguous, and therefore, communication always contains an element of uncertainty. To manage this uncertainty, human dialog partners continually monitor the conversation, their own comprehension, and the apparent comprehension of their interlocutor. Human partners elicit and provide feedback as the conversation continues, and make conversational adjustments as necessary. We contend that the ability to engage in this meta-dialog is the source of much of the flexibility displayed by humans when they engage in conversation; see (Perlis et al, 1998). We have demonstrated that enhancing existing dialog systems with a version of MCL that allows for meta-dialogic exchanges improves performance.

For instance, in one specific case tested, a user of the natural-language train-control simulation TRAINS-96 (Ferguson et al, 1996) tells the system to ``Send the Boston train to New York." If there is more than one train in Boston, the system may well choose the wrong one to send---the user may have in mind the train that runs regularly to and from Boston and so might respond: ``No, send the *Boston* train to New York!" Whereas the original TRAINS-96 dialog system responds to this apparently contradictory sequence of commands (Send, Don't send, Send) by once again sending the very same train, our MCL-enhanced version of TRAINS notes the anomaly (i.e., the contradiction in commands) and, by assessing the problem, identifies a possible explanation in its choice of referent for ``the Boston train". The enhanced system then chooses a different train the second time around, or if there are no other trains in Boston, it will ask the user to specify the train by name. The details of the implementation, as well as a specific account of the reasoning required for each of these steps, can be found in (Traum et al, 1999).

Our current dialog system, ALFRED, uses the MCL approach to resolve a broad class of dialog anomalies. The system establishes and monitors a set of dialog expectations related to time, content and feedback. For example, in a toy-train domain, if the user says ``Send the Metro to Boston", ALFRED notices that it doesn't know the word `Metro' (a failure of the expectation that it will find input words in its dictionary). Alfred's first response is to try to determine what it can about the unknown word. Since Alfred knows about the command ``send" and its possible arguments, it is able to determine that ``Metro" is a train. If it cannot determine from this which train the user is referring to, it will request specific help from the user, saying: ``Which train is `Metro'?" Once the user tells the system that `Metro' is another word for `Metroliner', it is able to correctly implement the user's request. See (Josyula 2005).

A fuller test of the RAH hypothesis

The above examples of MCL at work are well and good. But the larger promise of the method is that of a single domain-independent MCL rather than specially designed ones for each application; see (Anderson et al, 2007a,b; Schmill et al, 2007). Toward that end, we are about to embark on a more ambitious project, in which an upgraded MCL will simultaneously be applied to three distinct domains: NLP, virtual reconnaissance robots sending secure messages in a virtual "AfghanWorld", and physical robots exploring a physical mock-up of the Martian surface. There are then three broad kinds of agents here: an upgraded ALFRED for the NLP system, AfghanWorld security-sensitive robots, and Mars-World exploration robots. A human will communicate with ALFRED in English, and ALFED will—when appropriate—translate and forward commands to the various robots and also receive and tranlate into English robotic replies. Each agent type will be enhanced with the very same MCL code, in a deliberate attempt to assess MCL's adequacy to highly distinct agents and tasks.

And still fuller

Even if the above three-pronged study above is wildly successful, much more remains to be done. Fully general RAH should be also able to become host-and-domain *specific* over time, like the involuntary immigrant. But for this to occur, an agent will need substantial infrastructure. For instance, as already stated, it will need a range of trainable modules as well as appropriate training algorithms for them. It will also need a well-organized memory so that it can assess its adaptations over long time-periods, as well as progress on short-term tasks. With these and other additions, MCL might "fuse" with its "body" (system) and become one unified agent, akin to a baby's brain getting familiar with the baby's body. But our discussion so far has glossed over a major issue that will form a central part of this next phase of work: inference.

It may have occurred to the reader that, despite much mention early on of rationality and reasoning, little has been said here about how inference fits into our vision, other than the need for some version of reasoning that treats contradictions "responsibly". In fact, we have just such a formalism (active logic) and a reasoning engine based on it. Active logic addresses two key needs here: it not only recognizes (direct) contradictions, but it also has a real-time (i.e., evolving in real time) notion of what time it is *Now* – as inference goes on. Interestingly, the latter is key to the former: anomalies often make themselves known in the form *Expect(P)* and *Observe(-P)* at a particular time *t*, and this affords the logic engine the

option of inferring at time *t+1* that such an anomaly has occurred and is to be treated as such.

Both active logic and more traditional logics (even nonmonotonic ones) usually represent these as *P* and *–P*: one tends to believe that things will be as expected, and one also tends to believe ones senses. Belief revision treats the former as already part of a given logical theory, and the latter as a newcomer to be factored smoothly into the former by means of judicious excisions to preserve consistency.

But often one simply does not know whether to trust the newcomer observation over the existing expectation: further input might be required to make that call, and also it might not be important to adjudicate between them at all if they are not critical to one's concerns. Hence the need to a logic that *notices* such a contradiction (at time *t*) and that is wary enough (at times subsequent to *t*) not to foolishly trust both contradictands (unlike monotonic logics), but able that is then able to reason about (*assess*) their importance, and if needed then *guide* one or more possible resolutions into place. See (Anderson et al, 2004).

Thus active logic is central to the theme here. Nevertheless, most of our work on MCL to date has employed active logic mostly as a conceptual motivation, but not built in as an actual inference engine. Our next planned phase of development will include an active-logic engine as part and parcel of MCL, especially in the *assess* stage.

Conclusion

We have presented an approach to the brittleness problem motivated by considerations from biology and psychology. Several examples of implementations based on this idea were discussed, and directions for next steps were suggested. If this line of investigation holds to its promise, automated flexible cognition may be closer than we think.

References

Anderson, M. L., Chong, Y., Josyula, D., Oates, T., Perlis, D., Schmill, M., & Wright, D. (2007). Ontologies for Human-Level Metacognition. Proceedings of the 8th International Symposium on Logical Formalizations of Commonsense Reasoning. Menlo Park, CA: AAAI Press.

Anderson, M. L., , M. L., Schmill, M., Oates, T., Perlis, D., Josyula, D., Wright, D., & Wilson S. (2007). Toward Domain-Neutral Human-Level Metacognition. Papers from the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning. Menlo Park, CA: AAAI Press.

Anderson, M. L., Oates, T., Chong, Y., Perlis, D. (2006). The metacognitive loop: Enhancing reinforcement learning with metacognitive monitoring and control for improved perturbation tolerance. Journal of Experimental and Theoretical Artificial Intelligence.

Anderson, M. L., Oates, T., Perlis, D. (2006). ReGiKAT: (Meta)-Reason-Guided Knowledge Acquisition and Transfer --or Why Deep Blue can't play checkers, and why today's smart systems aren't smart. Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris.

Anderson, M. L., Josyula, D., Perlis, D., & Purang, K. (2004). Active Logic for more effective human-computer interaction and other commonsense applications", Proceedings of the Workshop for Empirically Successful First-Order Reasoning, International Joint Conference on Automated Reasoning.

Ferguson, G. M., Allen, J. F.,. Miller, B. W., & Ringger, E. K. (1996). The design and implementation of the TRAINS-96 system: A prototype mixed-initiative planning assistant. Technical Report 96-5, University of Rochester.

Josyula, D. (2005). A Unified Theory of Acting and Agency for a Universal Interfacing Agent. PhD Thesis, University of Maryland.

Perlis, D., Purang, K., Andersen, C. (1998). Conversational Adequacy: Mistakes are the essence, International Journal of Human Computer Studies.

Schmill, M., Josyula, D., Anderson, M. L., Wilson, S., Oates, T., Perlis, D., & Fults, S. (2007). Ontologies for Reasoning about Failures in AI Systems. Proceedings from the Workshop on Metareasoning in Agent Based Systems at the Sixth International Joint Conference on Autonomous Agents and Multiagent Sytems.

Schmill, M., Oates, T., Josyula, D., Anderson, M. L., Perlis, D., Wilson, S., & Fults, S. (2008). The Role of Metacognition in Robust AI Systems: AI Magazine. Menlo Park, CA: AAAI Press.

Traum, D., Andersen, C. F., Chong, Y., Josyula, D., Okamoto, Y., Purang, K., , O'Donovan-Anderson, M. L., & Perlis, D. (1999). Representations of Dialogue State for Domain and Task Independent Meta-dialogue: Electronic Transactions on Artificial Intelligence.