Incorporating Elements of a Processual Self into Active Logic

Justin Brody
Goucher College
Towson, MD

Michael T. Cox and **Donald Perlis**
University of Maryland Institute for Advanced Computer Studies
College Park, MD

Abstract

In a previous paper, we discussed the desirability of a formal model of a self that emerged when a process became immediately aware of its own processing. In this paper we discuss some of the features that go into a realization of this processual self in active logic. A description is given of some specific extensions to active logic that would move toward allowing for a computational processual self.

Processual Self

In (Brody, Cox, and Perlis 2013), we argued for the notion of an immediate processual self as a uniting theme across cognitive sciences. We have in mind a dynamic self that arises from a process immediately observing itself and responding to its own processing. An intuition for this kind of process can be developed by first imagining a program which records its own history and modifies its behavior based on that observation. A modification of this picture in which the observation and modification occur simultaneously describes the kind of process we are positing [1].

Such a self seems to underlie a vast array of cognitive phenomena, and consequently a formal model of a processual self has the potential to play a key role in approaching numerous puzzles. These range on topics from the question of how an agent is capable making references to questions about the relationship between mind and brain.

We made some very preliminary gestures toward how one might go about implementing such a self in a computational system. In this paper we take those ideas further and flesh out some of the components that might go into the creation of an immediate processual self.

[1] On its face the notion of an observation occurring at the same time as a response to that observation seems inherently paradoxical. We will skirt this paradox by introducing a thick notion of time.

It is interesting to compare this with the prima facie paradoxicality that is associated with the Liar Paradox in its various manifestations. In the case of Gödelian self-reference, we have a single sentence in which the predicate and the parameter are in a sense the same. More specifically, the parameter is a different object from the predicate but is interpreted as referring to the predicate. Similarly, in what follows we will have discuss forms of reference that are really between one moment and a previous moment but that are interpreted as being simultaneous.

Towards an Implementation

Active Logics

Active logics, described in (Anderson et al. 2002), (Anderson et al. 2008) and (Purang 2001), among other places, are a family of time-situated logics in which reasoning occurs in time and about time. It has several distinguishing features, including paraconsistency (the ability to work with contradictory sentences in a knowledge base). We discuss some of the features that make active logic a suitable basis for a formalism that can model an immediate processual self.

In the first place, active logics are *active*. That is, they do not view inference as a timeless phenomenon in which all the consequences of a knowledge base are immediately known to an agent. Rather, inference is carried out in time via a series of one-step deductions (single applications of an inference rule). This makes active logics appropriate for a processual model of self, since they inherently model reasoning and description as processes.

In the second place, active logics are *temporal*. Specifically, every inference occurs at a given time and is marked with a time-stamp. In particular, an active logic agent's inference of P at time t will be recorded in its database as $t : P$ Moreover, active logic has a built-in indexical *Now(t)* which holds precisely at moment t. In this way, active logics have a basic access to their own current state. Our plan is to extend this mechanism to allow for access to a wider portion of the current state.

Extending Active Logic

Our goal in this paper is to discuss extensions to active logic that would facilitate the implementation of an immediate processual self. Besides incorporating a wider notion of access to state, we also want to work with what Nicholas Humphrey has termed "thick time" (Humphrey 2006). In particular, there are two potential ways of conceiving the immediacy of a processual self. One could take as immediate something that occurs at the current instant – this is the usual notion of a zero-duration slice of time found in physics. This however doesn't allow enough time for a process to actually arise. For many purposes, then, we choose to work with

moments rather than instants. A moment will be an interval of times, viewed as an equivalence class. There are different approaches one could take to defining the equivalence classes. A simple approach would be to simply make every moment consist of an equal number of times (e.g. every 20th time-stamp would mark a new moment)[2]. Another approach would be to have moments centered on a particular content, so that a moment is the duration of time it takes to think a particular thought and have any associated meta-processing with it. We will not settle on a particular approach in this paper.

In our formalism, we will replace the time-stamps of active logic with time-stamp, moment, state triples (t, m, σ). Here the state is a tuple representing any extra data that an agent will keep track and make use of.

We want our agents to have capabilities of introspection, and would like our logic to have a certain kind of self-reference. We first observe that introspection can occur on different levels. One can be generally aware of what one is thinking about but also be more finely aware of the inner workings of one's thought process. This is reflected below in the distinction between the $PreviousThought$ and $Introspect$ predicates.

For self-reference, we contend that what arises phenomenally as (instantaneously) simultaneous self-reference is (at least potentially) often really momentarily simultaneous other-reference. That is, it is reference that occurs in a single moment, but if we divide that moment into instants it would actually be reference from a thought at one instant to a thought at another instant within the same moment. We can thus distinguish between *instantaneous self-reference* in which a logical term refers to itself at that instant, and *momentary self-reference* in which a term refers to another term in the same moment. Both kinds of self-reference will play a role in our discussion.

In our conception, the state σ will mirror the processing determined by the inference rules. We thus imagine a reflective inference engine, which can see and modify its own inferring in real-time. The thick moments make this possible – an inference process can use an understanding of its current behavior as a stimulus to modify its behavior in a single thick moment.

To implement this in active logic, we will need to specify what information the state variables contain, describe new inference rules which will determine the meaning of the new predicates we add to the active logic framework, and finally discuss the needed modifications to the inference engine itself. Many of these details will vary; we specify some possibilities that could form the core of a specification.

State The information tracked in the state component will vary. We give some possible components here (we expect that the first two of these will be present in any instantiation.)

CurrentInference: Inference in active logic proceeds by a series of 1-step deductions. That is a particular inference rule is applied to a set of antecedents in the knowledge base to derive a particular conclusion. The $CurrentInference$ is a triple (R, A, P) consisting of the inference rule, the set of antecedents, and the consequent of the application of that rule at the current time.

CurrentThought: The consequent of the $CurrentInference$

MomentaryThought: If we conceive of moments as constituting a particular thought and the auxiliary reasoning that goes with it, then this will be the content of that main thought.

Context: The context in which the current thought-process is occurring. This will generally be a complex data structure that will contain things like the goals that the agent is trying to accomplish, any time constraints it has, and other essential information about the current context.

Inference Our new inference rules will introduce several indexicals to supply access to the current state, and also some predicates to deal with actions (many implementations will probably want observation predicates as well). We give the indexical predicates first.

- *Now(t)*: This basic indexical is inherited from active logic and tracks the current time. Its semantics are determined by the rule

$$\frac{(t, m', \sigma') : \varnothing}{(t+1, m, \sigma) : Now(t+1)}$$

- *ThisMoment(m)*: Tracks the current moment in an analogous way. The semantics are determined by

$$\frac{(t, m, \sigma') : \neg NewMoment(t, m)}{(t+1, m, \sigma) : ThisMoment(m)}$$

- *PreviousThought(T)*: T is the thought that was just inferred.

$$\frac{(t, m, \sigma') : \sigma.CurrentThought == T}{(t+1, m, \sigma) : PreviousThought(T)}$$

- *Introspect(P)*: This corresponds to an agent observing itself at a given time-stamp, and is governed by the following inference rule:

$$\frac{(t, m, \sigma) : \exists R, A(\sigma.CurrentInference == (R, A, P))}{(t, m, \sigma) : Intropsect(P)}$$

While this particular predicate doesn't actually accomplish anything, it does demonstrate the possibility for instantaneous self-reference. Note in particular that P will always be $\ulcorner Introspect \urcorner$.

The discussion of actions is beyond the scope of the current paper. However, we envision that most agents will have mechanisms for such. In particular, any agent should have a predicate *Action(A)* so that when *Action(A)* is inferred, the inference engine will call the appropriate module to initiate the action specified by A. Similarly, an *Observe(O)* predicate should cause the inference engine to enter O into the knowledge base.

[2]There is a body of scientific literature on *temporal framing* that indicates that at least some parts of the human brain work this way. See especially (Gevins et al. 1983) and (Varela et al. 1981)

Another action we would like our agent to have is the ability to modify its inference rules. In particular, if we define an inference rule as a triple (R, A, C) where R is a name for the rule, A is a set of antecedents, and C is a conclusion, then we would want the following actions:

AddInferenceRule(R,A,C)

RemoveInferenceRule(R)

Engine The workings of the inference engine will mostly be the same as those of the standard active logic inference engine (e.g., see (Purang 2001) and (Josyula 2005)). We will, however, want the following extensions:

At every new time-step, the entire state should be updated to reflect the state of the system. While an inference rule was given above for the current time, this only works under the assumption that the underlying inference engine correctly increments the time counter at each step. We similarly need the engine to keep the state tuple current at each time-step.

Similarly, we need a mechanism for adding predicates *NewMoment(m)* to the knowledge-base and updating the moment counter. Note that it is not sufficient to specify an inference rule for this, since an active logic inference may not be applied immediately.[3]

If the conclusion of an inference is an action, then execute a module that performs that action. Similarly, if an observation comes in, it should be entered in the knowledge base at an appropriate time.

Since our history-tracking and self-reflection mechanisms are capable of drawing an infinite number of conclusions from a single moment, we will need practical cut offs on the amount of information generated. We could, for example, set a limit on the amount of nesting of information that could occur (with respect to self-referential statements).

The engine needs to be able to modify itself through the addition and deletion of inference rules in accordance with the actions given above.

Examples

Basic Existential Awareness

We imagine an agent that is simply aware of its own flow through time, without have any particular content to its thoughts. In particular, an agent with the following as its only inference rule

$$\frac{(t, m, \sigma) : \emptyset}{(t+1, m, \sigma) : Introspect(\ulcorner Introspect \urcorner)}$$

would simply examine its own state in perpetuity.

[3]It is a feature of active logics that they have 1-step deduction engines. In particular, between time t and time $t+1$ an active logic agent will only apply one of the inference rules available to it. As a consequence, an inference rule specifying, say, that at time 20 the moment counter should be incremented might not actually be applied until time 24.

I was Speaking in French, but now I'm speaking in English

We could also imagine an agent that is speaking to another agent in French. At some point, the agent switches to English for reasons which it is not conscious of (perhaps it has a hidden inference rule which tells it to do so at time s). Upon realizing this, the agent might say something along the lines of "I was speaking in French, but now I'm speaking in English". We imagine that it is in the very act of saying this sentence that agent decides how to complete it; beginning by saying "I was speaking in French", the agent realizes it isn't any longer and completes the sentence with "but now I'm speaking in English".

In particular, let us imagine that our agent keeps track of the current and previous speaking languages in its state, and that it has an action *Utter(S)* which causes it to utter the string S. If the agent was speaking in French, then it may have a token of the form

$$(t_0, m_0, \sigma_0) : (\sigma_0.CurrentLanguage == French)$$

During the current moment (m_1), perhaps the agent realizes that it has switched languages.

To implement this, we will use an expanded state that we will represent by a 6 tuple

$$(CurrentInference, CurrentThought, \\ MomentaryThought, Context, \\ PreviousLanguage, CurrentLanguage)$$

.

We describe a sequence of deductions that captures the described scenario. For space considerations we make a number of abbreviations (and omit times and moments; everything given can be assumed to take place in a single moment). Our notation is as follows:

- We indicate omitted state components with $\cdot$.
- C represents the context "Communicating with Agent2".
- F and E denote French and English, respectively.
- U abbreviates $Utter$
- We let S_1 denote the concatenation of the string "I was speaking in " with $\sigma.PreviousLanguage$, where σ is the current state.
- PL and CL denote the $PreviousLanguage$ and $CurrentLanguage$ components of the state.
- LC denotes the predicate $LanguageChanged$
- We let S_2 denote the concatenation of "but now I'm speaking in " with $\sigma.CurrentLanguage$.

Then the following sequence of deductions represents the agent's realization that it has switched languages and response to that realization.

State	Derived Formula
$(\cdot, \cdot, \cdot, C, F, E)$	$Action(\ulcorner U(S_1) \urcorner)$
$((\alpha, \sigma.L \neq \sigma.PL, LC), \cdot, \cdot, C, F, E)$	LC
$((\beta, LC, \cdot), \cdot, \cdot, C, F, E)$	$Action(\ulcorner U(S_2) \urcorner)$

Note that the self-reference here is momentary. In particular, within the system the reference is really other-reference; it only becomes self-reference when the sequence of times comprising the moment are taken as a single moment.

Modus Tollens

Finally, we imagine an agent that goes through its history and examines the conclusions it has drawn. It realizes at some point that whenever $P \rightarrow Q$ was in the knowledge base and P was as well, then it always eventually added Q to the knowledge base. It thus infers that is has a rule for modus ponens. Through further reflection, it adds a rule for modus tollens as well.

This requires some more sophisticated elements. In particular, we need mechanisms to parse elements of the knowledge base and a mechanism for drawing universal inferences.

We will therefore assume that we have a predicate $Implication(t, \ulcorner P \urcorner, \ulcorner Q \urcorner)$ with semantics that will hold whenever at time t an implication of the form $P \rightarrow Q$ is in the knowledge base (here P and Q can represent complex expressions as well as atoms).

We will also assume inductive rules for deriving universal statements. In particular, rather than needing to prove universals from first principles as in classical logic, our agent will draw universal conclusions based on it's experience. One of the great advantages of the non-monotonicity of active logic is that the agent can always retract such conclusions when confronted with a counterexample.

Specifically, we will introduce an inference rule that derives universals from the absence of a counterexample:

$$\frac{\bigwedge_{t'<t}(t', m', \sigma') : \neg\exists a \neg P(a)}{(t, m, \sigma) : \forall x P(x)}$$

As a practical consideration, it would make sense to only use this rule for fairly large values of t, so that the chance of having generated a counterexample is reasonable.

We also assume that knowledge is inherited: if the agent deduces P at time t then P remains in the knowledge base unless it is explicitly retracted.

Then for sufficiently large t, for any $t' < t$ the following statement will be provable for arbitrary P, Q

$$Implication(t', P, Q) \wedge P \rightarrow \exists s, m, \sigma\ (t + s, m, \sigma) : Q$$

Our inductive inference rule then suffices to conclude that this statement holds for all values of t, P and Q, establishing modus ponens as a statement in the knowledge base.

In order to derive *modus tollens* from *modus ponens*, our agent will need to reason about truth and have some form of basic epistemology. We introduce a predicate $True(\ulcorner P \urcorner)$ which will be used to represents the agent's beliefs about truth. We take two forms of the law of the excluded middle as axioms: $True(P \vee \neg P)$ and $True(\neg(P \wedge \neg P))$ (these are, of course, logically equivalent, but we're interested in developing an agent that can deduce basic logical laws for itself). We will assume that our inference engine supports the standard rules of first order natural deduction (including quantifier instantiation), although these needn't be encoded in the $Truth$ predicate.

Then, since it has just discovered the rule for *modus ponens* it can take this as a basic truth, adding $True((P \rightarrow Q \wedge P) \rightarrow Q)$ to its knowledge base. Because reasoning in active logic is non-monotonic, it can use its knowledge base as a kind of scratch area to reason about hypothetical situations. In particular, it can add assumptions to its knowledge base, reason about their consequences, retract all the information entered into the knowledge base since the original assumption, and finalize the result by entering a conditional statement into the knowledge base. This mirrors the process of proving conditionals via natural deduction.

To apply this to proving *modus tollens*, we can introduce a $Deduction$ context. In this context, the agent will understand its reasoning to be about general truths and will make derivations appropriate to that context. We can define inference rules for that context which will follow the pattern outlined above, and basically turn the agent's built-in reasoning into a natural deduction system. In particular, we note that any active logic agent has a need to specify how it will resolve contradictions. We can specify that in the $Deduction$ context contradictions serve to prove the negation of the appropriate assumption. We can also specify that the results of such a deduction should be encoded with the $Truth$ predicate.

A derivation of *modus tollens* could then proceed by a simple deduction: assuming both $(P \rightarrow Q) \wedge (\neg Q)$ and P leads to a contradiction, so that the agent can conclude $True\Big(\big((P \rightarrow Q) \wedge (\neg Q)\big) \rightarrow \neg P\Big)$.

Discussion

We have specified an extension to active logic that allows for various technical properties of an immediate processual self. We discuss the properties that are present and also where further work is needed. Our ultimate goal is the emergence of a dynamic self awareness that can ground a variety of cognitive phenomena, and our discussion will accordingly focus on what is needed for such an emergence. We first note that our system has a self-model in the form of its access to its current state. In fact, we argue that it has a self-model a very strong sense, since its model of itself is based on direct self-access, rather than being access to an intermediate representation. Similarly, the system is capable of generating the two kinds of self-reference mentioned above – instantaneous self-reference and momentary self-reference. One might naturally wonder whether any of the paradoxes that normally arise with self-reference are of concern here - in particular, is our system capable of generating any sentence analogous to "this sentence is false", and if so, is it capable of interpreting it coherently? If not, this may point to a potential weakness in our deductive capability. We intend to examine these questions further in future work. Finally, the system is capable of self-modification as demonstrated in our second and third examples. In particular, in response to its awareness of its own processing, an agent can modify its own inference rules and processing. One interesting question is whether our system is capable of strong self-reference. The distinction between strong and weak forms of self-reference is due to Perlis (Perlis 1997).

Weak self-reference occurs when a system's referring is not inherent in the system, but only occurs through the mediation of an outside agent. For example, if a random array of pixels line up in a way to produce an image of an arrow pointing to itself, then the arrow which appears is not intentionally referring to itself (or anything) – it requires an outside agent to observe the pixels and interpret them as doing so. Strong self-reference is self-reference that is not mediated in this way – the referring is self-referring without any external agent deeming it so. It is worth noting that the weak self-reference that seems to be inherent in computer systems has formed fodder for a number of philosophical arguments against functionalism (for example, see (Kripke 1982)). One of us has argued (Perlis 1997) that strong self- reference is required to have fully conscious agents. More strongly, it seems that it is required even for genuinely grounded reference to occur (Perlis 1991), (Perlis 1992). It would seem at first sight that the reference in any digital computer is inherently weak – the machine is simply a stream of electrical activity, it is only the agreed upon conventions between the programmers and the users that make a certain pattern of electrical activity a way of referring to the letter "x" (for example). In the case of self-reference, however, it may be possible to argue that a computer system examining its own state is genuinely doing so, especially when that examination is not mediated through an external representation. In particular, there is a causal connection between referent and referring in such a system that is much stronger than is normally present.

Ultimately, we would like to ask ourselves if the formalism presented here is capable of giving rise to an immediate processual self that is adequate to the heavy work we have claimed that such a notion can do. As things stand, it seems like a fair amount work needs to be done before this question can be answered. As a broad outline, we need to

1. Implement the ideas given here in a working system.
2. Specify in greater detail how a processual self helps solve various puzzles, and test our system against proposed solutions.
3. Understand the meta-logical properties of the enhanced active logic we are proposing here.

Thus a full reckoning of the utility of this system must wait for further progress. We do, however, believe that we have made a move in the right direction.

References

Anderson, M. L.; Josyula, D. P.; Okamoto, Y. A.; and Perlis, D. 2002. Time-situated agency: Active logic and intention formation. In *Workshop on Cognitive Agents, 25th German Conference on Artificial Intelligence*. Citeseer.

Anderson, M. L.; Gomaa, W.; Grant, J.; and Perlis, D. 2008. Active logic semantics for a single agent in a static world. *Artificial Intelligence* 172(8):1045–1063.

Brody, J.; Cox, M. T.; and Perlis, D. 2013. The processual self as cognitive unifier. In *Proceedings of the Annual Meeting of the International Association for Computing and Philosophy*.

Gevins, A. S.; Schaffer, R. E.; Doyle, J. C.; Cutillo, B. A.; Tannehill, R. S.; and Bressler, S. L. 1983. Shadows of thought- shifting lateralization of human brain electrical patterns during brief visuomotor task. *Science* 220(4592):97–99.

Humphrey, N. 2006. *Seeing red: a study in consciousness*. Harvard University Press.

Josyula, D. P. 2005. A unified theory of acting and agency for a universal interfacing agent.

Kripke, S. A. 1982. *Wittgenstein on rules and private language: An elementary exposition*. Harvard University Press.

Perlis, D. 1991. Putting one's foot in one's head–part i: Why. *Noûs* 435–455.

Perlis, D. 1992. Putting one's foot in one's head–part ii: How.

Perlis, D. 1997. Consciousness as self-function. *Journal of Consciousness Studies* 4(5-6):5–6.

Purang, K. 2001. Alma/carne: implementation of a time-situated meta-reasoner. In *Tools with Artificial Intelligence, Proceedings of the 13th International Conference on*, 103–110. IEEE.

Varela, F. J.; Toro, A.; Roy John, E.; and Schwartz, E. L. 1981. Perceptual framing and cortical alpha rhythm. *Neuropsychologia* 19(5):675–686.