# Electronic News Journal on Reasoning about Actions and Change

Editor: Erik Sandewall

Volume 2, Number 9

Contributions during September, 1998

ISSN, print: 1403 - 2031
on-line: 1403 - 204X

Linköping University Electronic Press
http://www.ep.liu.se/ej/enrac/1998/09/

# Publication Trail and Update Rules

The Electronic News Journals (ENJ) are a medium for exchange of scientific information and debate. In particular, they serve as the forum where articles received by the ETAI (Electronic Transactions on Artificial Intelligence) are discussed publicly for review, in separate News Journals for each ETAI area.

The Electronic News Journal on Reasoning about Actions and Change (ENRAC), in particular, uses a publication trail where contributed information is first distributed on a daily basis as *Newsletters* using electronic mail and web pages in HTML. Then, contributions during the same month are compiled into an HTML version of the monthly *News Journal*. Finally, the same information is converted via Latex to a postscript version that is suitable for printing on paper and reading off-line. It is formally published and archived by the Linköping University Electronic Press in both its electronic and its paper form, as described on page (iv).

**Offprints of individual segments.** Each News Journal issue consists of a sequence of *segments* relating to specific topics. "Offprints" of these can be obtained from the URL:s indicated at the head of the first page of each segment, or from the ENRAC service page specified below. It is intended to maintain these URL:s and their contents for the foreseeable future. Misprints of minor significance are corrected in these offprints (but not in the E-Press version since it is considered archival).

**Corresponding HTML edition with dynamic links.** The HTML edition contains considerable numbers of links to other pages and structures on the net: articles that are available on-line, home pages of conferences and of individual researchers, links to other part of the ETAI structure, and so on. Due to the natural limitations of the paper medium, only some of the WWW links have been retained here as footnotes. In order to retrieve articles and other information that are referenced in the present issue, it is recommended to look up the corresponding issue in HTML and to use its link. The HTML issues of the News Journals on Reasoning about Actions and Change can be found at the ENRAC *service page,*

`http://www.etaij.org/rac/nj/`

It is intended that the HTML issues will by updated continously to the largest extent possible, for example by replacing URL links to the home pages of authors that have changed to another site.

# Table of Contents

# Dates of Publication

Since the date of publication may be understood either as the date of first public appearance, or as the day of reproduction on paper in many copies, and since both of these definitions may be difficult to apply in the case of electronic publication, we make the following clarifying statement.

The contents of the present issue were put on-line in their original, versions as Newsletters in plaintext, HTML, and latex/postscript during the month of September, 1998. Then the contents were edited and formatted, resulting in the present, formatted version which was published on 16.7.1999, in two concurrent editions: an on-line edition and a paper edition. The on-line edition was timestamped electronically and put on-line by Linköping University Electronic Press at the URL specified on page (i). The paper edition was obtained by printing the on-line edition on a standard computer printer. It was reproduced in 25 copies, legally archived, and made available for distribution.

Vol. 2: *297-297*, 30.9.1998
http://www.etaij.org/rac/nj/9809/

# The Month of September

## Summary of News Journals Contents

Two articles were received in September: by Graham White on Simulation, Ramification, and Linear Logic, and by Michael Thielscher on the Fluent Calculus.

Linear logic has been proposed as a new and quite non-standard approach to reasoning about actions and change, and in particular Graham presented an article on that topic in the workshop on commonsense reasoning in London in January. The workshop discussions were recorded and continued in the the present News Journal, and showed considerable interest but also divergence of opinions on the matter. Graham's ETAI article is likely to rekindle the debate.

Michael's article is the first contribution to the special category of *reference articles* for approaches to actions and change, in response to the call for contributions that we made in June. Several more contributions were promised for the beginning of October. We shall therefore return to this topic in the next issue of the News Journal.

The ENRAC contained extensive discussion of David Poole's article during September. The present issue also contains the summary of the article.

It is our policy to encourage authors to submit a summary of their articles, besides the conventional abstract. The summary is supposed to be more detailed and specific, and a length of about two ETAI journal pages is recommended. The idea is to facilitate for readers to get a good idea about the contributions of the article without having to read it in detail.

The biggest discussions during August were for the articles by Marc Denecker et al, and by Murray Shanahan. These discussions continued in September as well.

**Electronic News Journal on Reasoning about Actions and Change** 298
Vol. 2: *298-299*, 30.9.1998
http://www.etaij.org/ra/rac/008/

# David Poole:

# Decision Theory, the Situation Calculus and Conditional Plans

## Summary of the article

This paper shows how to use the situation calculus within a logic based on Bayesian decision theory. Within this logic we can axiomatize the dynamics of a domain (including stochastic frame axioms, and stochastic ramification axioms), how noisy sensors work, and the utility of final states. This paper defines the expected utility of conditional plans (containing sequential composition and conditioning on sensor values) given a representation of the domain.

The first part of this paper argues that many of the problems with representations of discrete actions have arisen because the primitive mechanisms in first order logic to handle uncertainty (namely disjunction and existential quantification) are inadequate to handle the choices (and their consequences) that an agent needs to make. For example, logics with no uncertainty, such as acyclic logic programs, seem to handle frame and ramification axioms unproblematically. We build on these acyclic logic programs, but handle uncertainty with more sophisticated tools. The paper argues that the appropriate representation for uncertainty for an artificial agent is Bayesian decision theory. In Bayesian decision theory all uncertainty is handled by probability and all preference by utility.

The representation that we use to combine logic and Bayesian decision theory is the Independent Choice Logic (ICL). The ICL grew (a) from work on abduction and default reasoning where we want to abduce to causes and use assumptions to make predictions from these, (b) from building rule-structured conditional probability tables in Bayesian networks, and (c) from work on different agents choosing assumptions modelled on game theory. The ICL is built from a choice space (a set of mutually exclusive alternatives, where an alternative is a set of ground atomic formulae, called atomic choices), and an acyclic logic program that gives the consequences of the atomic choices. The logic program can't have any clauses that imply an atomic choice. The semantics is defined in terms of possible worlds; there is possible world for every selection of one element from each alternative. A formula is true in a possible world if it it true in the (unique) stable model of logic program together with the atomic choices selected. The ICL is naturally abductive: the explanations of $g$ form a concise specification of the possible worlds in which $g$ is true.

In this paper all choices are made by nature. There is a probability distribution over the alternatives (a non-negative number associated with each atomic choice so the values for the atomic choices in an alternative sum to one). When there are finitely many alternatives, the probability of a world is the product of the probabilities of the atomic choices selected by the world (when there are infinitely many alternatives we use a measure over sets of worlds in the standard manner, but where the alternatives are unconditionally independent). The probability of a formula is the sum of the probabilities of the worlds in which it is true. There is a close relationship between the ground (variable-free) ICL and Bayesian networks; the elements of the body of a rule correspond to the parents in the Bayesian network. The ICL gives rule-structured conditional probability tables.

The paper shows how the situation calculus can be embedded in the ICL, where each possible world specifies a complete history; it specifies what is true in each situation. The actions that make up the situation are the actions of the agents (that is, the motor controls sent by the agent). Thus the agent can know what situation it is in, even if it doesn't know what is true in any situation. The ICL specifies a probability distribution over fluents in each situation.

We also allow for (noisy) sensors where the sensor value is some stochastic function of what is true in a situation. The agent can condition its actions on its sensor readings. The agent then adopts a conditional plan, where the conditions are sensor reading. A conditional plan selects a final situation in each possible world, where the situation selected depends on the sensor readings in that possible world.

Utility is axiomatized as any other fluent; we assume there is a unique utility for each situation in each possible world (the utility that is obtained if the agent finishes in that situation). This can be used to define an expected utility for each conditional plan. The expected utility of a conditional plan can be computed by abduction; we explain the final utility and the sensor readings in the plan. Each explanation has an associated probability and implies a final situation, and the utility of that situation. The expected utility can be obtained directly from the explanations by computing the expectation over the utilities.

The aim of this work is not to see how many different formalisms we can put together, but to see how few we can get away with. We use a very simple logic of acyclic logic programs, a simple way to handle uncertainty in terms of independent choices, and the standard situation calculus. This lets us define the expected utility of conditional plans. It is an open question as to how far we can build on this without compromising the simplicity of the formal foundations.

**Electronic News Journal on Reasoning about Actions and Change** 
Vol. 2: *300-316*, 30.9.1998
http://www.etaij.org/ra/rac/008/

Protocol of on-line discussion during September, 1998
about the following research article:

# David Poole

# Decision Theory, the Situation Calculus and Conditional Plans

## A1. **David Poole** (14.9)

Uffe Kjærulff writes:

> *The focus of the paper seems to be on representational issues, which is fine. But, being a Bayesian-network person, I would find it interesting to see how computations of expected utilities of plans are performed in the ICL framework. This issue is only touched upon very briefly in the paper, mentioning that it involves enumerating all possible cases. In my understanding that amounts to constructing the corresponding decision tree (which may be huge), and then computing the probability and utility for each leaf of the tree.*
>
> *So, my question boils down to: Is there an efficient computational scheme for ICL?*

Thanks for the question! First, ICL theories can be seen as representations of Bayesian networks; there is a local translation between groundings of the ICL theories and Bayesian networks. ICL theories (with only choices by nature) can be seen as first-order rule-based representations of Bayesian networks. Thus any Bayesian network algorithms can, in principle, be used for the ICL. There are two main computational advantages of the ICL approach. The first is that they can exploit logic-based reasoning techniques for probabilistic reasoning (although it turns out that once the link is realized, it isn't restricted to ICL theories). Secondly they give us an opportunity to exploit, not only the graphical independence of a Bayesian network, but also the contextual independence that i natural in the rule based representations. I will expand briefly on each of these here.

The ICL (with only choices by nature) can be seen as a way to write first-order Bayesian networks as rules. There is a local translation between ICL theories and Bayesian networks. Note that a Bayesian network has nothing to say about how a node depends on its parents. The ICL lets us write the conditional probabilities as a set of rules. (This is often a more compact representation than conditional probability tables when there are contextual independencies).

To translate an ICL theory into a Bayesian network: ground the theory (substituting ground terms for the free variables); the ground atoms are the nodes of the Bayesian network; the atoms in the bodes of the rules that imply a become the parents of a. There is a lot more structure that can be expressed by the rules than in the network (in particular the contextual independencies). Note that the acyclicity of the logic program ensures that the resulting network is acyclic and (even when the resulting network is infinite) there are only finitely many ancestors for any node (which means you only need a finite subset of the graph to answer any conditional probability statement).

To translate a Bayesian network to an ICL theory, you write as rules how the parents of a influence a. The atomic choices are introduced as extra conditions on the rules to let us interpret the rules logically. For more details see (Poole 1993).

The first way that the ICL can be used computationally is adapting the logic-based algorithms for probabilistic reasoning. In particular, the algorithms for consistency-based diagnosis (such as the focusing algorithms of de Kleer), based on the notions of conflicts, can be adapted to compute posterior probabilities in the ICL (and so for Bayesian networks). This works well when there are skewed probability distributions. This has been explored in (Poole 1996).

Another approach is to exploit the structure that can be expressed in a Bayesian network as well as the rule structure of the ICL. This is ongoing work with Nevin Zhang. I'll first sketch how the efficient structured Bayesian network algorithms (such as clique-tree propagation and more query-oriented algorithms such as D'Ambrosio's SPI, Zhang and Poole's VE or Dechter's bucket elimination) can be seen as ways to exploit the factorization of the joint probability distribution. The basic operation is to sum out a (non-observed, non-query) variable. Algorithms such as clique-tree propagation are efficient because they can compile this operation into an secondary structure (the clique tree or join tree) that works by message passing in the join tree so that the posterior distribution for all variables can be obtained in twice the time it takes to compute the posterior on one. The basic operation of the message passing from one clique to another is to sum out the variable that appears in one clique that doesn't appear in the other.

It turns out that you can carry out a similar operation on the rule base directly. We have been working on a rule-based variation of the query-oriented algorithms I call probabilistic partial evaluation (Poole 1997). Eliminating a variable turns out to be like resolution. When the rules give no structure beyond that of the Bayesian network (i.e., there is no contextual independence, and there is a rule for each assignment of variables to a parent), the algorithm reduces to the standard query-oriented Bayesian network algorithms. But when there is contextual independence (as there is in many ICL theories) the algorithm lets us exploit that. This also seems to be very promising for approximation, where different approximations can be used in different contexts (Poole 1998).

Aside: One main difference between clique tree propagation and the query-oriented algorithms is that clique tree propagation acts on marginal distributions (each clique represents the marginal distribution of the variables in the clique), whereas the query-oriented algorithms act on conditional probabilities. Probabilistic partial evaluation relies on manipulating conditional probabilities. The contextual independencies get lost quickly when you reason with marginal distributions.

Finally, there is also a promise of structured dynamic programming using the rules. The most comment methods for solving sequential decision problems is to do dynamic programming. Traditional dynamic programming explicitly reasons over state space.s and thus suffers from what Bellman calls the "curse of dimensionality". We should be able to do much better by reasoning in terms of the variables (or propositions) that define the state space and exploiting the rule-structure in dynamic programming. In the ICL, dynamic programming looks much like regression planning. We can either do this in POMDPs where we regress values through conditional plans (Boutilier and Poole 1996) or where we are building plans conditional on history (as in influence diagrams), and we can use the rule structure to determine states that all have the same utility (Poole 1995).

Thus in answer to your question, this is an active area of research by myself and others who are looking at exploiting contextual independence for probabilistic inference and sequential decision making. One of the reasons I am optimistic that this will work is that it promises nice methods of approximation, where we can approximate differently in different contexts. We will have to see if we can deliver on this promise.

## References

The following are references to some of my papers that give more details (the HTML version gives hyperlinks to the actual papers):

C. Boutilier and D. Poole, "Computing Optimal Policies for Partially Observable Decision Processes using Compact Representations", Proc. Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland, Oregon, August 1996.

D. Poole, "Probabilistic Horn abduction and Bayesian networks", Artificial Intelligence, 64(1), 81-129, 1993.

D. Poole, "Exploiting the Rule Structure for Decision Making within the Independent Choice Logic", Proceedings of the Eleventh Conference on Uncertainty in AI, Montreal, August 1995, 454-463.

D. Poole, "Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks", Artificial Intelligence, 88, 69-100, 1996.

D. Poole, "Probabilistic Partial Evaluation: Exploiting rule structure in probabilistic inference", Proc. Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, August 1997, pp. 1284-1291.

D. Poole, "Context-specific approximation in probabilistic inference", Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98), Madison, Wisconsin, July 1998.

N.L. Zhang and D. Poole, "Exploiting Causal Independence in Bayesian Network Inference", Journal of Artificial Intelligence Research, 5, 301-328, 1996.

## Q2. **Ray Reiter** (11.9)

David,

I enjoyed reading your paper, and learned a lot from it. I agree with you that axiomatic action theories need an account of probability/decision theory, although I do not share your reservations about being able to do so in the sitcalc (see below). My comments and questions concern only the probabilistic (as opposed to the decision theoretic) components of your paper.

### Technical issues

1. Examples 2.8 and 2.9.

The predicate $pickup\_succeeds(S)$ in the first clause.

(a) It is not parameterized by the thing – in this case key – being picked up. It should be $pickup\_succeeds(key, S)$. Presumably, you need one of these predicates for each action type, e.g. *putdown_succeeds*, *goto_succeeds*, etc. Therefore, why not introduce a generic predicate $succeeds(action, S)$, in which case, you can write $succeeds(pickup(key), S)$, $succeeds(goto(loc), S)$, etc? Similarly, you can have a relation $fails(action, S)$. Now, $C_0$ contains $union(succeeds(pickup(key), S), fails(pickup(key), S))$, where the union is over all ground instances of these atoms. Incidentally, your specifications of what sets are in $C_0$, e.g. bottom p. 13, is an abuse of notation, since it looks like an object level sentence, when in fact it is a metalevel statement for which universal quantification over situations is not defined.

(b) In axiomatizing at on p. 20, you do not include a *goto_succeeds* atom in the clause body. Does the atom $would\_not\_fall\_down\_stairs(S)$ play this role?

2. In the conditional plan of p. 21, don't you want $at(key, r101)$ instead of *at_key*?

3. Consider the axiom of Example 2.9 and your comment "Note that this implies that putting down the key always succeeds". Suppose not. Then is it the case that the axiom should be modified by replacing the atom $A \neq putdown(key)$ by something like

$$\neg(A = putdown(key) \wedge putdown\text{-}succeeds(key, S))$$

If so, perhaps this should be pointed out. (See my call below for a guide for the perplexed.)

4. In the axiom of Example 2.9, you include a choice $keeps\_carrying(key, S)$, which you introduce because "there may be a probability that the agent will drop the key". You then axiomatize this with $P0(keeps\_carrying(key, S)) = .95$ But I can't think of any realistic way to axiomatize this probability when only the parameter $S$ is available.

Basically, you want to describe the probability that the robot doesn't lose the key during the interval between the occurrence of the last action in $S$, when $carrying(key, S)$ was true, and the occurrence of the next action $A$. In other words, you want to describe the probability that the robot is still carrying the key at situation $do(A, S)$ given it is carrying it in situation $S$. But that depends, at the very least, on WHEN this next action $A$ occurs. Now on one account of adding time to the sitcalc (Reiter, Proc KR'96), an action can take an additional temporal argument denoting that action's occurrence time, for example, $openDoor(t)$. Then one expects that $P0(keeps\_carrying(key, S))$ given that $openDoor(10)$ is the next action should be greater than this probability given that $openDoor(1000)$is the next action. So as far as I can tell, the choice $keeps\_carrying(key, S)$ should take another argument, namely $A$, or perhaps simply $time(A)$, where $time(A)$ is the occurrence time of the next action $A$.

## The need for action preconditions

On p. 8, you claim "None of our representations assume that actions have preconditions; all actions can be attempted at any time." I'm sceptical of this claim. Consider your example domain, where you adopt the axioms $P0(pickup\_succeeds(S)) = .88$ Just prior to this, you say "$P0(pickup\_succeeds(S))$ reflects how likely it is that the agent succeeds in carrying the key *given that it was at the same position as the key and attempted to pick it up*". So you can't seriously write $P0(pickup\_succeeds(S)) = .88$ because here $S$ is any situation, including one in which the agent is 1000 kms from the key. I take it that what you intended to write was something like

$$at(key, Pos, S) \wedge at(robot, Pos, S) \rightarrow P0(pickup\_succeeds(S)) = .88(*)$$

If this is the case, then it is exactly here where you are using action preconditions, namely, in characterizing those situations in which an action has a high probability of succeeding. It also appears that you are appealing to action preconditions in the axiom of Example 2.8 (namely, $at(robot, Pos, S) \wedge at(key, Pos, S)$). Probably, these atoms should be omitted because their effects are realized in the above axiom (*).

Finally, you often talk about actions being "attempted", as in the above quotation. I found this particularly confusing since nothing in your ontology deals with action "attempts". Actions simply realize their effects with certain probabilities. You do not, for example, have an action *try_pickup*, as do other authors (e.g. James Allan) who do try to axiomatize action *attempts.* In this connection, I had a hard time making sense of footnote 9, since it motivates the axiom of Example 2.9 in terms of talk about "trying" to pick up the key. In fact, even after reading this material several times, I couldn't figure out footnote 9 until, much later, I began to think about how your approach compares with Bacchus, Halpern and Levesque. See my comments below on BHL.

## Section 3.3

I must confess that I couldn't make much sense of the discussion in Section 3.3, especially your arguments that the sitcalc is inherently unsuitable for modeling multiple, possibly reactive agents. It's true that Golog lacks the right control structures for doing this, but your comments on p. 26 and p. 27 (p. 26 "In order for a situation calculus program to ... and does the appropriate concurrent actions." p. 27 "When there are multiple agents ... or complex actions.") suggest to me that you are not aware of more recent work by De Giacomo, Lesperance and Levesque on Congolog that addresses precisely these issues (Proc. IJCAI 97). Congolog provides reactive rules (interrupts) and interleaving concurrency in a semantically clean way, all formulated within the sitcalc. I believe that Congolog addresses all your criticisms of the sitcalc, except, of course, it does not (yet) incorporate utilities.

## Comparison with Bacchus, Halpern and Levesque

So far as I know, there have only been two proposals for augmenting the sitcalc with probabilities, yours and BHL. As I indicate below, these two approaches are considerably different, and therefore I believe it is important to elaborate on these differences more than you do (top p. 25) especially the substantial ontological differences.

For BHL, there is no *action_succeeds* and *action_fails* fluents. Instead, an action whose outcome depend on nature is represented by the nondeterministic choice of simpler, deterministic actions corresponding to each different outcome. For example, the nondeterministic action *flipaCoin* is represented as the complex action

$$flipaCoin = flipHeads | flipTails$$

where | is GOLOG's operator for nondeterministically choosing between the two actions *flipHeads* and *flipTails*. For your ongoing example, *pickup*(*key*) would be represented by

$$pickup(key) = pickup\_succeeds(key) | pickup\_fails(key)$$

When an agent executes the action *pickup*(*key*) in situation $S$, nature steps in and selects which of the two actions *pickup_succeeds*(*key*) and *pickup_fails*(*key*) actually occurs, and it does so with frequencies determined by an axiom of the form:

$$P0(pickup\_succeeds(key), S) = p \leftrightarrow (suitable\ conditions\ on\ S\ and\ p)$$

In other words, your atomic choice of bottom p 13, and your axiom for $P0$ top of p 14 are represented by BHL in the way I have indicated above. These are very different ontological and representational commitments than yours, and I think they deserve a deeper analysis than your discussion provides. Here are a few important differences these commitments lead to:

1. Primitive actions for BHL are entirely different than yours. BHL primitive actions would be things like *pickup_succeeds*(*key*), *flipHeads* etc, i.e. they consist of the deterministic actions that nature can perform, together with those deterministic actions (if any) that the agent can perform. For you, primitive actions are typically nondeterministic, and correspond to those actions that the agent can perform (but with nondeterministic outcomes chosen by nature), e.g. *pickup*(*key*), *flipaCoin*.

2. Therefore, for BHL, situation terms describe the "actual" history selected by nature in response to the agent performed actions, e.g.

$$do(pickup\_fails(key), do(flipHeads, S0)).$$

For you, situations denote the history of agent performed actions, e.g.

$$do(pickup(key), do(flipaCoin, S0)).$$

The BHL axioms all refer to the former situations. In your ontology, only the latter are allowed.

3. In BHL, all primitive actions (*pickup_succeeds*(*key*), *flipHeads*) are deterministic and successor state axioms (the causal laws + solution to the frame problem) are formulated only wrt these. For you, the causal axioms are formulated wrt the (nondeterministic) agent's actions (*pickup*(*key*), *flipaCoin*); hence the need for your fluents *pickup_succeeds*($S$) and *pickup_fails*($S$). These function as random variables that determine the effects of the nondeterministic action *pickup*(*key*). O ne consequence of this difference is that for BHL, successor state axioms are perfectly straightforward, whereas for you, because of these random variables, the causal laws become, at least conceptually, somewhat opaque (e.g. footnote 9). Moreover, BHL use

"classical" action precondition axioms, in contrast to you (see above about action preconditions).

4. For BHL, your nondeterministic agent actions like *pickup*(*key*), are not part of the language of the sitcalc. Rather, they are nondeterministic GOLOG programs. Since you also introduce a notion of a program (Sections 2.8, 2.10), there are some natural comparisons to be made. The obvious (and perhaps only) important difference is in the nature of the primitive program actions. For BHL programs, the primitives are the deterministic actions described above (e.g. *pickup_succeeds*(*key*), *flipHeads*) whereas for you, they are the agent actions like *pickup*(*key*).

## Suggestion: A Guide for the Perplexed

Your paper uses a single, ongoing example to illustrate the axiomatization style required by your approach. Personally, I couldn't abstract, from the example, a set of guidelines by which I could invent my own axiomatization for a different example of my choosing. I think it would considerably enhance the paper if you provided such guidelines. For example, for each action, tell the reader what axioms need to be specified. What choice of fluents needs to be made? (For example, there seem to be at least two categories of fluents – "normal" fluents like $carrying(X, S)$, and atomic choice fluents. How does one decide on the latter, when given the primitive actions and "normal" fluents of an application domain?) Similarly, for each fluent, what axioms need to be specified? Also, what needs to be axiomatized about the initial situation? Finally, what do all these axioms look like syntactically? In connection with this last question, notice that on p. 20, the clause with head $at(X, P, S)$ resembles neither an effect axiom (as in Example 2.8) nor a frame axiom (as in Example 2.9).

## A2. **David Poole** (18.9)

Ray,

Thanks for your comments. I will try to explain everything so that it makes sense, and try to explain why I am doing what I am doing. I'll answer the questions in order, although it may help to read the guide to the perplexed first.

## Technical issues

### 1. Examples 2.8 and 2.9.

*The predicate* $pickup\_succeeds(S)$ *in the first clause.*

*(a) It is not parameterized by the thing – in this case key – being picked up. It should be* $pickup\_succeeds(key, S)$.

What it says now is that the probability that picking up something (in this case a key) actually achieves the agent carrying the object doesn't depend on what is being picked up (nor does it depend on the position, as long as the robot and key are at the same position). You are right in that this probably isn't what you want. [This would really only make a difference if the key term was a variable.] As it stands now, $pickup\_succeeds(key, S)$ should probably be $pickup\_succeeds\_in\_carrying\_key\_when\_robot\_and\_key\_are\_at\_same\_position(S)$ .

*Presumably, you need one of these predicates for each action type, e.g. putdown_succeeds, goto_succeeds, etc. Therefore, why not introduce a generic predicate succeeds(action, $S$), in which case, you can write succeeds(pickup(key), $S$), succeeds(goto(loc), $S$), etc? Similarly, you can have a relation fails(action, $S$).*

In this case *succeeds* would not only be a function of the action and the situation, but also a function of the condition being achieved (in this case that the key is being carried), and a context (in this case that the robot and the key are at the same position). But this is all that the rules say. The rules should be seen as axioms of conditional effects (see below).

*Now, $C_0$ contains union(succeeds(pickup(key), $S$), fails(pickup(key), $S$)), where the union is over all ground instances of these atoms.*

I am not sure what you mean by the union, but $C0$ would contain infinitely many 2-element sets. (See the answer to the next point.)

*Incidentally, your specifications of what sets are in $C_0$, e.g. bottom p. 13, is an abuse of notation, since it looks like an object level sentence, when in fact it is a metalevel statement for which universal quantification over situations is not defined.*

Yes. I mean for each situation term $S$ the set
$\{pickup\_succeeds(S), pickup\_fails(S)\}$
is in $C0$ . That is, I am quantifying over terms in the language.

Here $C0$ is a set of 2-element sets of ground terms. In this case $C0$ is infinite but countable. I could have made it a set of open formulae, in which case it could be finite, but this would have complicated the presentation for no obvious gain.

*(b) In axiomatizing at on p. 20, you do not include a goto_succeeds atom in the clause body. Does the atom would_not_fall_down_stairs($S$) play this role?*

Yes.

*2. In the conditional plan of p. 21, don't you want at(key, r101) instead of at_key?*

No. *at_key* is the sensor defined in Example 2.11 that (noisily) senses whether the robot is at the same position as the key. The robot can only condition on its sensed values. The robot doesn't know if the key is at room r101; it only knows what its sensors tell it. [You could also imagine that a robot could also condition on internal state variables, and what can be computed from internal state variables, which is hinted at in Section 2.10.]

*3. Consider the axiom of Example 2.9 and your comment "Note that this implies that putting down the key always succeeds". Suppose not. Then is it the case that the axiom should be modified by replacing the atom $A \neq putdown(key)$ by something like*

$$\neg(A = putdown(key) \wedge putdown\text{-}succeeds(key, S))$$

*If so, perhaps this should be pointed out. (See my call below for a guide for the perplexed.)*

This is sort of correct, but it doesn't get the probabilities right. I would expect that the probability of keeping the key to be independent of of the other two cases. There are three free parameters to be assigned probabilities, so we need three choice alternatives (when we only have binary alternatives). See the guide for the perplexed below; I do this example in more detail.

*4. In the axiom of Example 2.9, you include a choice $keeps\_carrying(key, S)$, which you introduce because "there may be a probability that the agent will drop the key". You then axiomatize this with*

$$P0(keeps\_carrying(key, S)) = .95$$

*But I can't think of any realistic way to axiomatize this probability when only the parameter $S$ is available.*

*Basically, you want to describe the probability that the robot doesn't lose the key during the interval between the occurrence of the last action in $S$, when $carrying(key, S)$ was true, and the occurrence of the next action $A$. In other words, you want to describe the probability that the robot is still carrying the key at situation $do(A, S)$ given it is carrying it in situation $S$. But that depends, at the very least, on WHEN this next action $A$ occurs. Now on one account of adding time to the sitcalc (Reiter, Proc KR'96), an action can take an additional temporal argument denoting that action's occurrence time, for example, $openDoor(t)$. Then one expects that $P0(keeps\_carrying(key, S))$ given that $openDoor(10)$ is the next action should be greater than this probability given that $openDoor(1000)$ is the next action. So as far as I can tell, the choice $keeps\_carrying(key, S)$ should take another argument, namely $A$, or perhaps simply $time(A)$, where $time(A)$ is the occurrence time of the next action $A$.*

Yes. That is right. If the agent could drop the key at any time, we need to model (or learn) how the agent may drop the key as a function of time. Presumably this would be modelled as a Poisson process (assuming it is equally probable that the agent can drop the key at any time). As outlined on page 1, I haven't dealt here explicitly with time.

The axiomatization I gave assumes that the agent can hold on to the key tightly enough so whether it drops the key isn't a function of time.

## The need for action preconditions

*On p. 8, you claim "None of our representations assume that actions have preconditions; all actions can be attempted at any time." I'm sceptical of this claim. Consider your example domain, where you adopt the axioms $P0(pickup\_succeeds(S)) = .88$ Just prior to this, you say "$P0(pickup\_succeeds(S))$ reflects how likely it is that the agent succeeds in carrying the key* given that it was at the same position as the key and attempted to pick it up*". So you can't seriously write $P0(pickup\_succeeds(S)) = .88$ because here $S$ is any situation, including one in which the agent is 1000 kms from the key. I take it that what you intended to write was something like*

$$\begin{aligned} & at(key, Pos, S) \wedge at(robot, Pos, S) \rightarrow \\ & \quad P0(pickup\_succeeds(S)) = .88 \end{aligned} \qquad (*)$$

*If this is the case, then it is exactly here where you are using action preconditions, namely, in characterizing those situations in which an* *action has a high probability of succeeding. It also appears that you are appealing to action preconditions in the axiom of Example 2.8 (namely, $at(robot, Pos, S) \wedge at(key, Pos, S)$). Probably, these atoms should be omitted because their effects are realized in the above axiom (*).*

There are two separate issues here, first the meaning of preconditions, and secondly how the atomic choices act in contexts where they aren't applicable.

There are three things that could be meant by a precondition of an action:

*A condition under which the action can be done. A condition under which some effect follows from the action. A condition under which the action should be done.*

It is only the second, which are often called conditional effects, that I represent. The first is what I meant as "preconditions", which I don't have. (These correspond to your *Poss* predicate, if I am not mistaken). This shouldn't really be seen as a feature, but as a necessary evil. It means that I have to axiomatize the effect of an action under all conditions (including the conditions under which you may say the action is not possible). I do this because, in general, an agent may carry out an action even if it doesn't know that the action is "possible" (in the normal sense). The third point, where the action should be done, is obtained because I want to build sensible agent that maximize utility, but I also want to be able to evaluate how good arbitrary agents are, not just good ones.

The second issue is a bit more subtle. Let's assume that the choice alternatives are all binary (non-binary alternatives are useful when a relation is functional). In this case, there are the same number of alternatives as there are free parameters (conditional probabilities that can be independently assigned) in the probability model. For example, when converting a Bayesian network to an ICL theory, there are the same number of alternatives as there are numbers to be assigned in the Bayesian network. You typically have, for each effect $e$, rules of the form:

$e \leftarrow context_1 \wedge ac_1$

...

$e \leftarrow expcontext_k \wedge ac_k$

where the $context_i$'s are disjoint and covering. In this case we can assign the probabilities of the atomic choice $aci$ as the conditional probability:

$P0(aci) = P(e \mid context_i)$

Your question on conditions for $P0$ could be recast as: What happens to the $ac_i$ in the context when the corresponding rule is not applicable? Basically it gets marginalized out. We don't have to worry about it. The abductive characterization of the ICL shows that we only need to worry about the atomic choices that are needed to prove a goal (in our case the utility and the observations).

This should be seen in the context of an overriding theme of this work. We are not trying to see how much we can put into a formalism, but how little we can get away with. We don't need conditions on the probabilities of the atomic choices, so we don't have them.

*Finally, you often talk about actions being "attempted", as in the above quotation. I found this particularly confusing since nothing in your* *ontology deals with action "attempts". Actions simply realize their effects with certain probabilities. You do not, for example, have an action try_pickup, as do other authors (e.g. James Allan) who do try to axiomatize action attempts. In this connection, I had a hard time making sense of footnote 9, since it motivates the axiom of Example 2.9 in terms of talk about "trying" to pick up the key. In fact, even after reading this material several times, I couldn't figure out footnote 9 until, much later, I began to think about how your approach compares with Bacchus, Halpern and Levesque. See my comments below on BHL.*

Perhaps the term "attempted" is misleading. What I mean is that the robot sends the motor control for the action, irrespectively of whether the robot can do it (in this sense it is attempting the action). By the pickup(key) action I mean a particular motor control (the one that would achieve carrying the key if it is at the same position as the key and the pickup succeeded). This action can be carried out in any context; it just doesn't achieve carrying the key in these contexts.

## Section 3.3

*I must confess that I couldn't make much sense of the discussion in Section 3.3, especially your arguments that the sitcalc is inherently unsuitable for modeling multiple, possibly reactive agents. It's true that Golog lacks the right control structures for doing this, but your comments on p. 26 and p. 27 (p. 26 "In order for a situation calculus program to ... and does the appropriate concurrent actions." p. 27 "When there are multiple agents ... or complex actions.") suggest to me that you are not aware of more recent work by De Giacomo, Lesperance and Levesque on Congolog that addresses precisely these issues (Proc. IJCAI 97). Congolog provides reactive rules (interrupts) and interleaving concurrency in a semantically clean way, all formulated within the sitcalc. I believe that Congolog addresses all your criticisms of the sitcalc, except, of course, it does not (yet) incorporate utilities.*

"It's true that Golog lacks the right control structures for doing this..." I am quite happy with the agent having simple control structures. I am concerned about modelling asynchronous external actions or events (by other agents or by nature).

It is not clear to me that Congolog addresses this, in the sense that all of the concurrency in the language is internal to the agent. It doesn't model exogenous actions (they generated externally). This would seem to mean that external events are restricted to occur at the situations defined by the primitive events. [I am looking forward to being told I am wrong.]

Moreover, with respect to the mix of the situation calculus and time, it is clear now that we can do what we want when we have both the situation calculus and time. But when you have statements saying when an event occurred, you don't need situations. Thus parsimony would suggest that we do away with situations. [I would expect that this should form a different thread.]

## Comparison with Bacchus, Halpern and Levesque

*So far as I know, there have only been two proposals for augmenting the sitcalc with probabilities, yours and BHL. As I indicate below, these two approaches are considerably different, and therefore I believe it is important to elaborate on these differences more than you do (top p. 25) especially the substantial ontological differences.*

Yes, I do need to discuss this further (that is the joy of ETAI discussion of papers; we can retrospectively expand those parts that are of most interest).

First let me say that I am not claiming that mine is better than theirs, nor that it is different just for the sake of being different. We start from very different perspectives. In their introduction, they explicitly contrast their work with the work starting from Bayesian networks. In particular, they claim "...they [Bayesian networks] have difficulties in dealing with features like disjunction ...". I take a Bayesian perspective that disjunction is not a feature we want! What defines a Bayesian is that probability is a measure of belief, that any proposition can have a probability (both of which BHL would agree with) and that all uncertainty should be measured by probability. Thus you need to keep in mind that the different design choices could stem from this different perspective.

*For BHL, there is no action_succeeds and action_fails fluents. Instead, an action whose outcome depend on nature is represented by the nondeterministic choice of simpler, deterministic actions corresponding to each different outcome. For example, the nondeterministic action flipaCoin is represented as the complex action*

$$flipaCoin = flipHeads | flipTails$$

*where | is GOLOG's operator for nondeterministically choosing between the two actions flipHeads and flipTails. For your ongoing example, pickup(key) would be represented by*

$$pickup(key) = pickup\_succeeds(key) | pickup\_fails(key)$$

*When an agent executes the action pickup(key) in situation S, nature steps in and selects which of the two actions pickup_succeeds(key) and pickup_fails(key) actually occurs, and it does so with frequencies determined by an axiom of the form:*

$$P0(pickup\_succeeds(key), S) = p \leftrightarrow \\ (suitable\ conditions\ on\ S\ and\ p)$$

*In other words, your atomic choice of bottom p 13, and your axiom for P0 top of p 14 are represented by BHL in the way I have indicated above.*

Except that BHL don't model probabilistic actions (in their IJCAI-95 paper, which is the only one I had seen) although I believe they could. All of their probabilities are within the agent, so such axioms would have to be part of the formula for updating belief. [I just found a new paper, Reasoning about Noisy Sensors and Effectors in the Situation Calculus, 1998, at Faheim Bacchus's web site that does this. I'll call this BHL98.]

One other important point to note is that the probabilistic part of the ICL is a restricted form of Bacchus's and Halpern's logics of probability as belief. The translation is that each atom that isn't an atomic choice is defined by Clark's completion of the rules defining that atom. We need statements that the elements of an alternative are exclusive and covering and that the atomic choices that aren't in the same alternative are probabilistically independent. Thus the translation doesn't require "suitable conditions on $S$ and $p$".

A general framework for probability and action is intuitively straightforward (see Halpern and Tuttle (1993) for a general semantic framework). You can think about forward simulating the system. Nondeterministic/stochastic actions split the worlds and impose a probability over the resulting worlds. In the ICL we handle splitting the worlds using one simple mechanism: independent choice alternatives. I treat actions by the agent and other stochastic mechanisms (e.g., the ramifications of actions) in exactly the same way. BHL-98 split the worlds by nondeterministic actions (as you outline). They would then need a different mechanism for stochastic ramifications.

*These are very different ontological and representational commitments than yours, and I think they deserve a deeper analysis than your discussion provides. Here are a few important differences these commitments lead to:*

*1. Primitive actions for BHL are entirely different than yours. BHL primitive actions would be things like pickup_succeeds(key), flipHeads etc, i.e. they consist of the deterministic actions that nature can perform, together with those deterministic actions (if any) that the agent can perform. For you, primitive actions are typically nondeterministic, and correspond to those actions that the agent can perform (but with nondeterministic outcomes chosen by nature), e.g. pickup(key), flipaCoin.*

Yes.

*2. Therefore, for BHL, situation terms describe the "actual" history selected by nature in response to the agent performed actions, e.g.*

$$do(pickup\_fails(key), do(flipHeads, S0)).$$

*For you, situations denote the history of agent performed actions, e.g.*

$$do(pickup(key), do(flipaCoin, S0)).$$

*The BHL axioms all refer to the former situations. In your ontology, only the latter are allowed.*

Yes. I tried to be explicit about both of these points.

*3. In BHL, all primitive actions (pickup_succeeds(key), flipHeads) are deterministic and successor state axioms (the causal laws + solution to the frame problem) are formulated only wrt these. For you, the causal axioms are formulated wrt the (nondeterministic) agent's actions (pickup(key), flipaCoin); hence the need for your fluents pickup_succeeds($S$) and pickup_fails($S$). These function as random variables that determine the effects of the nondeterministic action pickup(key). One consequence of this difference is that for BHL, successor state axioms are perfectly straightforward, whereas for you, because of these random variables, the causal laws become, at least conceptually, somewhat opaque (e.g. footnote 9).*

Except I would think that my successor state axioms are perfectly straightforward. I get by with just Clark's completion, in much the same way as other logic programming representations of action. Frame and ramification axioms are treated the same (in fact I don't even think of them as different, and the paper doesn't distinguish them). I get by with Clark's completion because it is defined with respect to each world, where I just have a simple acyclic logic program. Maybe the guide to the perplexed below will help.

*Moreover, BHL use "classical" action precondition axioms, in contrast to you (see above about action preconditions).*

So what happens if the agent doesn't know if the precondition holds, but still wants to do the action? This is important because, with noisy sensors, agents know the actual truth values of virtually nothing outside of their internal state, and most preconditions of actions are properties of the world, not properties of the agent's beliefs.

*4. For BHL, your nondeterministic agent actions like pickup(key), are not part of the language of the sitcalc. Rather, they are nondeterministic GOLOG programs. Since you also introduce a notion of a program (Sections 2.8, 2.10), there are some natural comparisons to be made. The obvious (and perhaps only) important difference is in the nature of the primitive program actions. For BHL programs, the primitives are the deterministic actions described above (e.g. pickup_succeeds(key), flipHeads) whereas for you, they are the agent actions like pickup(key).*

One major difference is that in BHL, it is the agent who does probabilistic reasoning. In my framework, the agent doesn't (have to) do probabilistic reasoning, it only has to do the right thing (to borrow the title from Russell and Wefald's book). The role of the utility is to compare agents. This is important when we consider what Good calls type 2 rationality and Russell calls bounded rationality (see the work by I.J. Good, Eric Horvitz, and Stuart Russell for example), where we must take into account the time of the computation done by the agent in order to compute utility. I would expect that optimal agents wouldn't do (exact) probabilistic reasoning at all, because it is too hard! That is why we want a language that lets us model agents and their environment, and defines the expected utility of an agent in an environment. One of the reasons I didn't want to just add "time" into the framework is that I want a way to take into account the computation time of the agent (the thinking time as well as the acting time), and that makes it much more complicated (I wanted to get the foundations debugged first).

## Suggestion: A Guide for the Perplexed

*Your paper uses a single, ongoing example to illustrate the axiomatization style required by your approach. Personally, I couldn't abstract, from the example, a set of guidelines by which I could invent my own axiomatization for a different example of my choosing. I think it would considerably enhance the paper if you provided such guidelines. For example, for each action, tell the reader what axioms need to be specified. What choice of fluents needs to be made? (For example, there seem to be at least two categories of fluents – "normal" fluents like $carrying(X, S)$, and atomic choice fluents. How does one decide on the latter, when given the primitive actions and "normal" fluents of an application domain?) Similarly, for each fluent, what axioms need to be specified? Also, what needs to be axiomatized about the initial situation? Finally, what do all these axioms look like syntactically?*

Thanks for the suggestion. The following is a paraphrase of how we tell people to represent knowledge in Bayesian networks, translated into the language of the ICL.

First I'll do the propositional (ground) case. We totally order the propositions. The idea is that we will define each proposition in terms of its predecessors in the total ordering. For each proposition $e$, a parent context is a conjunction of literals made up of the predecessors of $e$, such that the other predecessors are conditionally independent of $e$ given the context. We find a set of mutually exclusive and covering set $\{context_1, ..., context_k\}$ of parent contexts. (The atoms appearing in one of the parent contexts are the parents of $e$ in the corresponding Bayesian network). If $e$ is always true when $context_i$ is true, we write the rule:

$$e \leftarrow context_i$$

If $P(e \mid context_i)$ isn't 0 or 1, we create a rule:

$$e \leftarrow context_i \wedge ac_i$$

and create an alternative $\{aci, naci\}$ (where these are both atoms that don't appear anywhere else), with the probability:

$$P0(ac_i) = P(e \mid context_i)$$

If the context is empty ( $e$ doesn't depend on any of its predecessors) we don't need to create a rule; we can just make $e$ an atomic choice. When the probability given the context is 0, we don't write any rules.

The case for the ICL with the situation calculus is similar. Intuitively, we make the total ordering of the propositions respect the temporal ordering of situations. We write how a fluent at one situation depends on fluents (lower in the ordering) at that situation and on fluents at previous situations. Note that this rule automatically handles ramifications as well as frame axioms. The total ordering of the fluents guarantees the acyclicity of the rule base and that we don't have circular definitions. The initial situation is handled as any other, but the predecessors in the total ordering are only initial values of fluents (and perhaps atoms that don't depend on the situation).

The only thing peculiar about this is that we often have fluents that depend on values at the current as well as the previous situation. This is important when there are correlated effects of an action; we can't just define each fluent in terms of fluents at the previous situation. But this also means that we don't treat frame axioms and ramification axioms differently.

Let's look at how we would do Examples 2.8 & 2.9, with putting-down the key possibly failing. Lets consider when carrying would be true. Suppose carrying doesn't depend on anything at the same time, so we can ask what are the contexts on which $carrying(O, do(A, S))$ depends. It would seem there are three such contexts:

> *$A = pickup(O)$, and the robot and O are at the same position.*
>
> *The action isn't $pickup(O)$, the robot was carrying O in situation S, and the action wasn't to put down O.*
>
> *The robot was carrying O in situation S, and the action was to put down O.*

In no other cases would the robot be carrying O.

The robot would be carrying O in the first context if the pickup succeeds (hence the name of the atomic choice in Example 2.8). This results in the rule:

$$\begin{aligned}&carrying(O, do(pickup(O), S)) \leftarrow\\&\quad at(robot, Pos, S) \wedge at(O, Pos, S) \wedge pickup\_succeeds(O, S)\end{aligned}$$

The robot would be carrying O in the second context if it keeps carrying the key (hence the name of the atomic choice in Example 2.9). This results in the rule:

$$\begin{aligned}&carrying(O, do(A, S)) \leftarrow\\&\quad carrying(O, S) \wedge A \ = putdown(O) \wedge\\&\quad A \ = pickup(O) \wedge pickup\_succeeds(O, S)\end{aligned}$$

The robot would be carrying O in the third context if the putdown failed. Thus, to account for the chance that the putdown failed, we would write the extra clause:

$$\begin{aligned}&carrying(O, do(putdown(O), S)) \leftarrow\\&\quad carrying(O, S) \wedge putdown\_fails(O, S)\end{aligned}$$

where $putdown\_fails(O, S)$ is an atomic choice, with $P0(putdown\_fails(O, S))$ the conditional probability that the robot is still carrying O after it has carries out the $putdown(O)$ action.

Note that putdown may succeed in doing other things, it just fails in stopping the robot from carrying O. So maybe *putdown_fails* should be something more like *putdown_fails_to_stop_the_robot_from_carrying*.

Footnote 9 is there to try to explain what the rules mean if the rule bodies aren't disjoint. We can interpret what these rules mean, but it usually isn't what you want (for the cases where both rules are applicable). When we write these rules, having non-disjoint rules is usually a bug (the implementation available from my web page warns when it finds non-disjoint rules.)

> *In connection with this last question, notice that on p. 20, the clause with head $at(X, P, S)$ resembles neither an effect axiom (as in Example 2.8) nor a frame axiom (as in Example 2.9).*

It is a ramification of the robot moving. The above methodology handles these ramifications in the same way it handles frame axioms (and any other axioms).

Thanks for your comments and questions. I hope this made the paper clearer. I would be happy to answer any other questions you may have (or engage in debate about the usefulness of this).

David

**Electronic News Journal on Reasoning about Actions and Change** 317
Vol. 2: *317-318*, 30.9.1998
http://www.etaij.org/ra/rac/009/

Protocol of on-line discussion during September, 1998
about the following research article:

# Marc Denecker, Daniele Theseider Dupré, and Kristof Van Belleghem

# An Inductive Definition Approach to Ramifications

## C2-5. **Eugenia Ternovskaia** (11.9)

Dear Marc, Daniele, and Kristof,

Thank you for an interesting discussion regarding your paper. It clarified some subtle points of your work and helped to understand your motivation.

> *An interesting question generated by your remark is whether Thielschers approach to derive causal rules from state constraints (about one state) and dependency information can be generalised to "constraints between states" (plus dependency information as far as needed).*

This is probably a question to Michael. I found his idea about generating some of causal rules interesting, and I am also curious whether it is possible to generalize it.

Best regards,
Eugenia

## C2-6. **Michael Thielscher** (16.9)

Dear Eugenia, Marc, Daniele, Christoph,

I'd like to second Eugenia's view on the "Counter" counter-example, which implies that the intended behavior can be obtained after all with my approach. The natural "inter-state constraint" in this example would be

$$\neg Holds(out, s) \wedge Holds(out, do(a, s)) \wedge Holds(count(n), s) \rightarrow Holds(count(n+1), do(a, s))$$

(in conjunction with $Holds(count(m), s) \wedge Holds(count(n), s) \rightarrow m = n$). Then if you consider the causal relationship

$$out \text{ causes } count(n+1) \text{ if } count(n)$$

(in conjunction with $count(n)$ causes $\neg count(m)$ if $m \neq n$), the only consistent successor state would show the expected increment of the counter by one. (A side remark: Inter-state constraints like the above can be expressed in a novel Situation Calculus-style formulation of the Fluent Calculus; if you're interested, have a look at *http://pikas.inf.tu-dresden.de/~mit/publications/conferences/JELIA98.ps)*

You have raised an interesting question in this context, namely, whether and how causal relationships can be extracted from such inter-state constraints. The current procedure is not directly applicable because it is based on formulas without situation or time argument. My feeling is that a generalization must be possible–there definitely is a formal correspondence between the above state constraint and the causal relationship–but working out the details will require some brain work.

Cheers.
Michael

Vol. 2: *319-323*, 30.9.1998
http://www.etaij.org/ra/rac/010/

Protocol of on-line discussion during September, 1998
about the following research article:

# Murray Shanahan

# A Logical Account of the Common Sense Informatic Situation for a Mobile Robot

## A2. Murray Shanahan (11.9)

Chitta,

Many thanks for your comments and questions.

> *(i) I think that in general a robot does not really sense events, rather it notices changes in certain fluent values (in say resistors linked to the sensors) from which it abduces the occurrences of triggered events. Nevertheless, it is ok with me to skip this part and directly talk about robots observing triggered events. But a clarification would help.*

But isn't a change in the voltage passing through a sensor an event? Especially if we consider that voltage passing through a threshold value.

> *(ii) Since you mention several times (abstract, Section 2, etc.) that you use a novel solution to the frame problem using 'Releases' it will be nice if you say about it a little more than what is said in Section 2 (just after EC 5). (You do use it in axioms in later sections, but you don't discuss them.)*

The use of "Releases" isn't really a big feature of the solution. More important is the division of the theory into parts which are circumscribed separately. This is what I call "forced separation" in my book, "Solving the Frame Problem" (MIT Press, 1997). There's a lot of detail about this solution to the frame problem in that book, which makes me a bit reluctant to duplicate it here, as it really is a side-issue. Maybe I should add a few sentences if things aren't clear. But perhaps the following answers will suffice.

> *Although from the illustrations I can appreciate the use of 'Releases' in formulating continuous change, its utility in formulating constraints is not clear to me. For example, why not eliminate B4 and instead of 'Releases' have 'Terminates' in the head of rule E2.*

Yes we could do that. But only because I've abstracted away the continuous motion in the Rotate action, and made it instantaneous. If the Rotate action initiated a period of continuous motion, then we would have to use Releases, as in the case of the Occupies fluent. This is because the axioms enforce the following rule:once a fluent has been initiated (terminated) directly by an event, only another event that directly terminates (initiates) it can change its value.

> *To me the advantage of having constraints such as B4 is that by having it we avoid explicit compilation of it to effect axioms. I.e. we can use constraints for automatically deducing effects indirectly. But if we have both B4 and E2 then we are not really avoiding the compilation, as E2 is like an effect axiom.*

Actually the purpose of (B4) here is to cut out spurious models. (But again, this only applies if we represent the continuous motion properly, instead of abstracting it away as I have done here.) Without (B4), during the period of continuous rotation of the robot, it could be facing in many different directions at once.

> *(iii) I am not sure if 'Bearing' is a standard geometrical notion. Perhaps an intuitive meaning of it will help. Or may you can use the more familiar notion of 'slope'.*

The word "bearing" is used in navigation. That's why I chose it. Maybe it's not the mathematically most appropriate term.

> *(iv) You say (just after Sp3): "the term Line(p1, p2) denotes the straight line whose ...". Perhaps it is more appropriate to say "straight line segment" instead of "straight line".*
>
> *(v) I think in axiom (B3) you are assuming velocity to be one. If you do please mention it.*
>
> *(vi) The sentence after (B6). In it you explain Blocked(w1,w2,r) by 'if object w1 cannot move any distance at all in direction r without overlapping with another object'. Perhaps you should say: "if object w1 cannot move any distance at all because of w2 in direction r without overlapping with another object'.*

I'll make the changes you suggest. Thanks.

> *(vii) When defining HoldsAt(Touching(w1,w2,p),t) are you making some assumptions about the shape of w1 and w2. Imagine two triangles which touch at a point, you can align them such that they touch but yet do not satisfy the conditions you have in B6.*

Yes, I think you're right. But the definition in (B6) is correct for the sense of touching required here, which excludes cases like the one you mention. I should probably add a sentence on this.

> *(viii) I am not clear about the intuition behind the notion of partial completion in Section 5; especially in the text after (5.7). For example in proposition 5.8, I would encode the intuition "bump switches are not tripped at any other time" by*
>
> $$Happens(Switch1, t) \rightarrow t = T_{bump}$$
> $$Happens(Switch2, t) \rightarrow t = T_{bump}$$
>
> *I am not clear about the intuition behind your formulation.*

I think your formulation is equivalent to mine. I just elected to separate the if and the only-if halves of the completion.

*Also, why not use the standard Clark's completion and explain the Clark's completion of $\Psi$, where $\Psi$ is as mentioned in Definition 5.9 and the paragraph before. I.e, have the Clark's completion of $\Psi$ in the right hand side of the turnstyle in Proposition 5.8*

I can't use the Clark completion, because I only want to complete the Happens predicate for sensor events. I might have incomplete knowledge about other events.

*(ix) In section 6 you logically express a region as a list of straight lines. (You say it just before Bo3.) Is there any particular reason you use lists instead of sets. If not, since you already use set notation in the rest of your formulation, by using sets here also, you might avoid additional axioms such as Bo3 and Bo4*

I don't have a good answer to this. I guess I could have used sets, and the result would have been slightly more succinct.

(x) Also (Bo3) is a fixpoint expression and you probably need (as needed when defining transitive closure) to minimize 'Members' to get the right models.

*(Recall that the logic program*

$$anc(X,Y) \leftarrow (X,Y)$$
$$anc(X,Y) \leftarrow (X,Z), anc(Z,Y)$$

*is not equivalent to the formula*

$$anc(X,Y) \leftrightarrow (X,Y) \vee ((X,Z) \wedge anc(Z,Y))$$

*)*

*You are not minimizing 'Members' in Proposition 6.2. Or perhaps I am missing something.*

There is an axiom missing, which says that nothing is a member of Nil. I'll put this in the final draft.

But given that, I don't think there's a problem here. It's easy to prove, for example, that $\neg Member(A, Cons(B, Cons(C, Nil)))$, just by recursing down the list (assuming A, B, and C are distinct). It's not the same as the transitive closure case.

*(xi) Although I can appreciate the usefulness of the logical formulation in Section 2-7, some may pose the following question: Why not just formulate as in Section 8 (with some extensions perhaps) and then have the correctness of the algorithm in Section 9 proven with respect to the formulation in Section 8. Why go through the formulation in Section 2-7? I think it will be a good idea to address this or say a few lines about this to preempt such questions/attacks on logical formulation.*

This is a very good question, and the issue is fundamental to our community's approach to AI. I don't think I can answer it fully here. But one argument would appeal to the fact that perception is a process which involves knowledge and reasoning, and if we're going to use logic for knowledge representation and reasoning in "higher level" cognitive processes, then we should use it throughout. I think this argument becomes more forceful when we consider less trivial sensors than the bump switches used here.

I hope that answers all your questions. Many thanks for taking so much trouble over my paper.

Murray

## Q3. **Paulo Eduoardo Santos** (11.9)

Thank you very much for answering my first questions. I really think it is a very interesting paper.

The second bunch of questions follows:

**1.** In the last answer set we read:

> *By splitting the circumscription into parts, we get a more managable theory, one whose consequences are easier to work out. In fact, the role of circumscription here is mainly to complete certain predicates, specifically Initiates, Terminates and Happens. If we circumscribe everything together, we have to do a lot more work to avoid difficulties like the*

I agree that Separation avoids the Hanks-McDermott problem , but doing so you get a restriction in the theory, since you are not able to write formulae using Initiates (or Terminates) and Happens together (on the contrary it won't be clear where to place such formulae in the circumscriptions). Thus, we are not able to express (and infer) some facts.

Then, why do not use negation-as-failure instead, since, for the particular case of circumscription we are interested in, the former is equivalent to the later (as proved in [j-aij-38-75]).

In which sense circumscription is more powerful than negation-as-failure in this framework?

**2.** I could not understand the last formula of page 14. There a biimplication is used defining the region $g$ with the formula $\Pi$. My interpretation of this is that $\Pi$ is not an approximation of $g$ but it describes exactly $g$. Is that what it means?

**3.** In the beginning of page 15 it is considered, as a solution to the abduction process, "conjunctions $M2$ of formulae" such that a given sensor data ($\Psi$) is logical consequence (considering the background theory).

My question is: what do you do if there is no $M2$ that explains $\Psi$ ? (I'm not considering the case where $M2$ follows from the background theory).

Many thanks for this opportunity,

Paulo Eduardo Santos

## References:

[j-aij-38-75] Michael Gelfond, Halina Przymusinska, and Teodor Przymusinski. *On the Relationship between Circumscription and Negation as Failure.* Artificial Intelligence Journal, vol. 38 (1989), pp. 75-94.

## Q4. **Paulo Eduoardo Santos** (17.9)

Dear Murray Shanahan,

As you suggested, I am sending those questions to ETAI:

1. On page 5 we read the definition of the following circumscription policy:

> *Given a conjunction of Happens and Initially formulae $N$, a conjunction of Initiates, Terminates and Releases formulae $E$ (...), we are interested in:*
>
> $CIRC[N; Happens] \wedge$
> $CIRC[E; Initiates, Terminate, Releases] \wedge U \wedge EC$

My question is: Why didn't you circumscribe *Initially* and *Happens* in parallel in the first conjunctive term of the expression above (since we should expect to have, as true in the time 0, only the explicitly stated facts and nothing else.)?

2. The second question concerns the same subject. On page 9 we have the predicate *AbSpace* defined in terms of *Initially*:

$AbSpace(w) \leftarrow Initially(Occupies(w, g))$

Further on we read:

> *The predicate AbSpace needs to be minimised, with Initially allowed to vary.*

The circumscription policy w.r.t. this idea is described on the top of page 10 :

$CIRC[O \wedge M; AbSpace; Initially] \wedge ...$

Question: Why do we have to circumscribe *Abspace* letting *Initially* vary, and not circumscribe only *Initially* (or both *Initially* and *Abspace*)?

Cheers,

Paulo

**Electronic News Journal on Reasoning about Actions and Change** 324
Vol. 2: *324-324*, 30.9.1998
http://www.etaij.org/ra/rac/012/

# Graham White:

# Simulation, Ramification, and Linear Logic

## Abstract of the article

This article first argues that formalisations of the "frame problem" should have certain desirable logical features; it then proposes a treatment of the frame problem, using linear logic together with modal operators, which fulfils these *desiderata* and seems to be successful in other respects.

**Electronic News Journal on Reasoning about Actions and Change** 325
Vol. 2: *325-325*, 30.9.1998
http://www.etaij.org/ra/rac/013/

# Michael Thielscher:

# Introduction to the Fluent Calculus

## Abstract of the reference article

The present introduction to the Fluent Calculus is intended as an ETAI reference article. It summarizes basic definitions and concepts in the Fluent Calculus, and is intended as a reference for future articles where the calculus is used.