Electronic News Journal on Reasoning about Actions and Change

Editor: Erik Sandewall

Volume 1, Number 4

Contributions during November, 1997

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

Linköping University Electronic Press
http://www.ep.liu.se/ej/enrac/1997/04/

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. 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 following URL:

`http://www.ida.liu.se/ext/etai/actions/njl/`

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, HTML version during the month of November, 1997. Then the contents were edited and formatted, resulting in the present, formatted version which was published on June 10, 1998, 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 200 copies, legally archived, and made available for distribution.

Vol. 1: *117-117*, 30.11.1997
http://www.ida.liu.se/ext/etai/actions/nj/9711/

The Month of November

Summary of News Journal Contents

The major event during this month was that Paolo Liberatore's article "The Complexity of the Language $\mathcal{A}$" was accepted to the ETAI. This was ETAI's first paper acceptance; it followed after due ETAI procedure that includes both an on-line open discussion (*reviewing*, in ETAI terminology) and a subsequent *refereeing* decision by confidential referees. The first ETAI issue (Vol. 1, Nr. 1-3) contains both Liberatore's article and an editorial note explaining the reviewing procedure in terms of this first case.

The month obtained an even mix of activities in the ENRAC Colloquium. There were contributions to the review debates about two other ETAI submitted articles in addition to Liberatore's. The two "panel" discussions received a number of contributions, many of which centered around the usefulness and relevance of action description languages such as $\mathcal{A}$ and $\mathcal{E}$, as compared to the direct use of first-order logic.

November, 1997 can therefore be identified as the month when both the ETAI publishing scheme and the ENRAC's role in relation to the ETAI became concretely operational: all major aspects of the scheme can now be demonstrated in terms of *how they work*, and not merely in terms of how we intend them to work.

The relationship between our research area and its applications, actual and potential ones, had been mentioned in a letter by Austin Tate in the September, 1997 issue. The same question was again brought up in the context of a CFP for a journal special issue - see page 156 in the present issue.

Some minor changes have been made to the graphical appearance of the latex/ postscript version of the News Journal for November, compared to the previous months. The major difference is that we go to a smaller font size.

Electronic News Journal on Reasoning about Actions and Change 118
Vol. 1: *118-121*, 30.11.1997
http://www.ida.liu.se/ext/etai/received/actions/001/

Protocol of on-line discussion during November, 1997
about the following research article:

Paolo Liberatore:

The Complexity of the Language $\mathcal{A}$

Q5. **Anonymous reviewer** (21.11)

Section 6 in the article addresses domain descriptions in which some states are not reachable from the initial state. I think the modification of the semantics of $\mathcal{A}$ as proposed in that section does not allow to solve the problem of unreachable states. Consider the situation where you replace the observation that H and L are initially both false by what you observe after two steps, i.e. replace the two 'initially' propositions by

H after $I; I$
$\neg L$ after $I; I$

Intuitively, the model should be the same. But the reachable states semantics now considers $\{H, L\}$ to be a possible initial state. This is because $\Psi_D(I, \{H, L\})$ is undefined there, hence both value propositions are weakly true. I.o.w., you get back to the orginal $\mathcal{A}$ semantics you wanted to avoid.

Generally, my feeling is that in order to solve the problem that your example highlights (and I think it is a problem) you cannot do in the language of $\mathcal{A}$. What you need is a notion such as "action A is executable if ...".

Also, on page 5 you write:

"to prove that D entails $\ulcorner F$ after $A_1; ...; A_m \urcorner$ it suffices to prove that $\ulcorner D \cup \neg F$ after $A_1; ...; A_m \urcorner$ is inconsistent."

This reduction seems not to go through for your modified semantics of section 6.

The rest (i.e. the main part of the paper) is just perfect as far as I can see. And it was a pleasure to read the paper.

A5. **Paolo Liberatore** (24.11)

You wrote:

> *Section 6 in the article addresses domain descriptions in which some states are not reachable from the initial state. I think the modification of the semantics of $\mathcal{A}$ as proposed in that section does not allow to solve the problem of unreachable states. Consider ...*

You are right. In this case, the semantics does not capture the intuitive meaning of the domain description.

You also wrote:

> *Generally, my feeling is that in order to solve the problem that your example highlights (and I think it is a problem) you cannot do in the language of $\mathcal{A}$. What you need is a notion such as "action A is executable if ...".*

The intended semantics for the actions in $\mathcal{A}$ is: all actions are executable in any state. When $\Psi_D(I,\sigma)$ is undefined, the choice of the original semantics of $\mathcal{A}$ is that the whole domain description is inconsistent. What I meant is instead that A is not executable in σ. In this sense, in $\mathcal{A}$ *it is possible* to express propositions like "I is executable if", but only in a non-intuitive way: in the example of the counter I is not executable in $\{H, L\}$. We need an appropriate semantics for such propositions.

(If there is a proposition "I is executable if ..." there is no need to make Ψ defined on I in states in which I is not executable. As a result, there is nothing that prevent us from using the indefiniteness of Ψ to infer the non-executability of actions. The drawback is that this definition is less intuitive.)

Analyzing the problem of unreachable states, my first idea for solving it was: an interpreted structure (σ, Φ) is a model of D if and only if

1. each action is executable in any state reachable from σ
2. each value proposition is satisfied (as in the old semantics of $\mathcal{A}$)

However, I discarded this idea, and instead I defined the one that is presented in the paper. Note that this semantics, although correct wrt your example, suffers from some drawbacks. For example, if for any state there is an action not executable there, then the domain description is inconsistent. Consider for example the case

I is executable if F
J is executable if $\neg F$

This domain is inconsistent, because in the initial state in which F is true the action J is not executable, while in the initial state in which F is false the action I is not executable. In such cases, it seems possible to infer that, if the initial state is $\{F\}$ then the first action executed is I. For instance, if there is also a proposition

initially F

the first executed action is I. The semantics of $\mathcal{A}$ does not infer any statement about the execution of actions (in $\mathcal{E}$ and similar languages there are propositions like "I at 0" or "I happens at 0"). A statement like "F after I" may be interpreted in two ways:

1. if F *were* executed the result would be a state that implies F
2. F is executable in the initial state, and the resulting state implies F

(The choice of the semantics of the paper is the first one, while the discarded one uses the second one.) Statements like "I at 0" or "I happens at 0" means that I is executed in the time point 0 (and thus the action must be executable in that state).

I discarded the first semantics (preferring the one of the paper) because the impossibility of executing an action in the initial state (or any other state) may influence the set of the possible initial states. The semantics of the paper first determines the set of possible initial states, and only then determines which states are reachable from them. However, in the example of the counter, the initial state $\{H, L\}$ must be rejected *because* the action I is not executable from there. This is why the semantics of the paper fails.

Perhaps the discarded solution is better then the chosen one. The problem is left open. A minimal requirement for a semantics that takes into account the reachability of states is that if D does not have "executable" propositions (or, if Ψ_D is total) then the new semantics must coincide with G&L's semantics. The semantics of the paper (and the one of the previous paragraph) has this property (clearly, this is not enough.)

You also wrote:

> *Also, on page 5 you write:*
>
> > "to prove that D entails $\ulcorner F$ after $A_1; ...; A_m\urcorner$ it suffices to prove that $\ulcorner D \cup \neg F$ after $A_1; ...; A_m\urcorner$ is inconsistent."
>
> *This reduction seems not to go through for your modified semantics of section 6.*

Indeed, this property holds only for the classical semantics of $\mathcal{A}$ (this property is used for proving the complexity of the entailment in the classical semantics of $\mathcal{A}$).

Q6. **Thomas Drakengren** (24.11)

Paolo,

Here are some questions and suggestions about details in your article:

page 16, line -14: What about the case A causes G, A causes $\neg G$? That would be inconsistent. What is the intended meaning in this case?

page 17, line 8: This case is similar to the IJCAI '97 paper by myself and Marcus Bjäreland [c-ijcai-97-1447], where we require one negative precondition and one negative postcondition (which is of course equivalent, replacing true and false). You can probably add some nondeterminism here, retaining tractability, if you do not allow a precondition, the same way as we're doing it (you can then have a Horn postcondition).

page 19, line 3: An inconsistent domain description could never entail the same propositions as a consistent one. What is the intended meaning in this case?

References:

[c-ijcai-97-1447] Thomas Drakengren and Marcus Bjäreland. *Reasoning about Action in Polynomial Time.* Proc. International Joint Conference on Artificial Intelligence, 1997, pp. 1447-1453.

A6. **Paolo Liberatore** (1.12)

Dear Thomas,

Thank you for your suggestions. About your observations:

> *page 16, line -14: What about the case A causes G, A causes $\neg G$? That would be inconsistent. What do you mean in this case?*

In this case Ψ_D is not total. This can be verified in polynomial time (Lemma 1).

> *page 19, line 3: An inconsistent domain description could never entail the same propositions as a consistent one! What do you mean in this case?*

The original domain description implies (under the semantics of reachable states) the same propositions of the modified one (using Gelfond & Lifshitz's semantics). The point is that I am using two entailment relations for the original domain description and modified one.

Paolo

Electronic News Journal on Reasoning about Actions and Change 122
Vol. 1: *122-122*, 30.11.1997
http://www.ida.liu.se/ext/etai/received/actions/002/

Protocol of on-line discussion during November, 1997
about the following research article:

Michael Thielscher

A Theory of Dynamic Diagnosis

Q1. **Rob Miller** (3.11)

Michael, a question regarding the relative likelihoods of different components failing:

It seems to me that, in the absence of specific domain knowledge to the contrary, it's right to prefer explanations of system failure which are minimal in terms of the number of failures of individual components, other things being equal (this is one aspect of how your main example works, if I've understood correctly). But you also mention that good diagnosis is able to "take into account a priori knowledge of differences in the likelihood of components to break" (an example might be the knowledge that relays are more likely to fail than resistors). Could you say a little more about this in relation to your approach? Would you perhaps use more than one "ab" predicate, with some sort of prioritised minimisation between the predicates?

A1. **Michael Thielscher** (3.11)

Rob,

My theory does indeed respect domain knowledge of the kind you mention. This knowledge is formally represented by a partial ordering among the instances of the (unique) *ab* predicate–which is equivalent to your suggestion of allowing a priority hierarchy of different *ab* predicates. The notion of preferred models (Definition 10) reflects a given partial ordering in that more unlikely instances of *ab* are preferably minimized. As a consequence, the axiomatization of the theory by means of the Fluent Calculus (Section 5) employs Brewka's prioritized extension to classical Default Logic.

Vol. 1: *123-130*, 30.11.1997
http://www.ida.liu.se/ext/etai/received/actions/003/

Protocol of on-line discussion during November, 1997
about the following research article:

Antonis Kakas and Rob Miller:

Reasoning about Actions, Narratives and Ramification

C2-1. **Alessandro Provetti** (10.11)

Dear Antonis and Rob,

I'd like to comment on Tom's example about the role of h-statements in $\mathcal{A}$-languages. In the language $\mathcal{L}$ of Baral et al. the theories:

F at $S0$

and

F at $S0$
A occurs-at $S0$

have different models.

Assume that there are other fluents than F, the former has models which differ one to another by the interpretation of the initial state (except -of course- for F) while agreeing on the fact that nothing happened at all.

The latter yields the same models as far as the initial state is concerned, *but* all of them sanction that A has happened. As a result, the latter theory implies the formula A occurs-at $S0$.

It appears to me that the equivalence of the two theories above under $\mathcal{E}$-semantics does not mean in general that $\mathcal{A}$-style semantics cannot account for h-propositions.

You may want to comment on this in the paper or -possibly- proceed to work on the entailment associated to $\mathcal{E}$.

Hope this helps. Ciao!

Alessandro Provetti

C2-2. **Tom Costello** (11.11)

Dear Antonis and Rob, (and Alessandro)

While the languages $\mathcal{L}_0$ and $\mathcal{L}_1$ of Baral et al. give truth conditions for propositions stating happens , precedes and holds , they do not give truth conditions for causes propositions. Like Baral and Gelfond, and Kartha and Lifschitz their models are functions from sequences of actions to (sets of) states. Because of this they conflate domain descriptions that are not conflated by models that are functions from states to sets of states (Res etc.).

Consider the following domain description, stated in $\mathcal{A}$.

A causes F
initially F

or in $\mathcal{L}_0$

A causes F
F at $S0$

These have the same functions from sequences of actions to sets of states as,

A causes F
A causes G if not F
initially F

or in $\mathcal{L}_0$

A causes F
A causes G if not F
F at $S0$

However, if we consider functions from states to sets of states, then these have different models. Thus domain descriptions that were distinguished by $\mathcal{A}$, are conflated by later languages.

These later approaches conflate models that intuitively differ.

I agree with Alessandro that $\mathcal{E}$ type languages can give semantics to h-propositions. My complaint is that current approaches fail to give semantics to all their propositions. As $\mathcal{A}$ and $\mathcal{E}$ type languages do not have a proof theory, save by being translated into other approaches, it seems strange that they do not even have a model theory for all their propositions. In Antonis and Rob's case they lack truth conditions for some of their propositions, and worse, it seems that it is not even possible to define truth conditions. The same problem arises for causal statements in Baral et al., Baral and Gelfond, and Kartha and Lifschitz. Other models of $\mathcal{A}$ type languages do not have this problem of collapsing domain descriptions $\mathcal{A}$ considered distinct, for causal statements, for instance, E. Giunchiglia, N. Kartha and V. Lifschitz, "Representing action: indeterminacy and ramifications". Therefore, I argue that action language models should define truth conditions for all their propositions, and further, should ensure that intuitively different models are distinct.

Tom

C2-3. **The Authors** (12.11)

Hi Tom,

You wrote:

In Antonis and Rob's case they lack truth conditions for some of their propositions, and worse, it seems that it is not even possible to define truth conditions.

As we said in our original answer to your question, it's trivial to extend the semantics of the Language $\mathcal{E}$ to include truth conditions for h- and c-propositions, but superfluous (to the main themes of the present and previous papers). However, for the record, you can do this by defining an interpretation as a tuple $\langle H, J, K\rangle$. H is as before, J is a function

$$\textsf{Actions} \times \textsf{Time-points} \longrightarrow \{ \textsf{true} , \textsf{false} \}$$

and K is a function

$$\textsf{Actions} \times \textsf{Fluent-literals} \times 2^{\textsf{Fluent-literals}} \longrightarrow \{ \textsf{true} , \textsf{false} \}$$

The definition of a model is exactly as before (Definition 9), with the additional conditions:

- $J(A,T) =$ true iff $\ulcorner A$ happens-at $T\urcorner$ is in D
- $K(A,L,C) =$ true iff either $\ulcorner A$ initiates L when $C\urcorner$ is in D, or $L = -F$ and $\ulcorner A$ terminates F when $C\urcorner$ is in D.

But this doesn't really add much insight; you just get that D entails a given h- or c-proposition iff the proposition is in D. Of course, for other extensions of $\mathcal{E}$ it might become worthwhile complicating the structure of an interpretation in this way. (Similarly for r-propositions.) Again, you might find Van Belleghem, Deneker and Dupre interesting in this respect.

Rob and Tony

C2-4. **Tom Costello** (13.11)

Dear Rob and Tony,

Your definition of truth for c-propositions seems very unintuitive to me. I would think that if A terminates F if G, then A terminates F if G, H.

Your definition does not give this result. The reason I ask for truth conditions for your propositions is that I cannot understand what the intuitive consequences of a set of propositions should be, unless I understand what the propositions say. If the propositions are expressed in a standard logic, then I understand them using the definition of truth in a model. However, your propositions are not in a standard logic, and therefore, to understand what

A terminates F if G

means, I have to know when it is true.

Your paper introduces a new type of proposition,

F whenever $G_1, ..., G_n$

There are some obvious choices for truth conditions for this type of proposition. In particular, it can be understood that every this is obeyed at every time-point, or that this is a property of every "possible" state, not every "actual" state. Without knowing which notion this proposition is trying to express, I cannot understand what the proposition says.

I do not think truth conditions are a side point to the main theme of your paper. As you say, action languages are supposed to be "understandable and intuitive". Languages cannot be understood without semantics.

Yours,

Tom

C2-5. **The Authors** (17.11)

Tom,

We think that perhaps we're in danger of going round in circles in this discussion. As we've said in other answers, we've much sympathy for your stance on the benefits of general purpose logics (and in particular classical logic), and that's why we've stated on numerous occcasions that languages such as $\mathcal{E}$ are perhaps best regarded as intermediate stages in the development of formalisms written in such logics. However, we do feel that they have a use in initially discussing and illustrating approaches to particular issues - in our case, to ramifications - in a relatively intuitive and uncluttered way. But we do recognise that what is intuitive for one person might not be so for another. (In particular, of course, as regards formalising common sense it is possible to supply classical logic axiomatisations which are intuitive to some people but not others).

Again, it would be interesting to get some views from more people who have developed $\mathcal{A}$ style languages on some of the general issues that you've raised (if not here, then perhaps in a more general ENRAC panel discussion on the advantages and disadvantages of specialised action languages).

Rob and Tony

> **Editor's note:** *continued discussion on the merits and demerits of Action Description Languages will be referred to the panel discussion on ontologies.*

C2-6. **The Authors** (28.11)

Tom,

In ENRAC 21.11, in the context of the general discussion on action description languages, you asked:

> *Similarly, does*
>
> $\mathsf{Always}\ F, G$
>
> *or Kakas and Miller's*
>
> $F\ \mathsf{whenever}\ \neg G$
>
> *mean that every actual state satisfies F,G, or every possible state.*

In the light of this remark, it now occurs to us that a possible partial explanation of your difficulty in gaining an intuition about the meaning of $\mathcal{E}$'s c- and r-propositions is that you're thinking in terms of states and state-transitions (natural enough if one is used to working with the Situation Calculus and related formalisms). But $\mathcal{E}$'s vocabulary and underlying ontology doesn't include (global) states - just fluents, actions and time-points. So it's difficult for us to see what you might be refering to by a "possible state" in the context of $\mathcal{E}$.

To understand our intentions, it's better to think just in terms of local cause and effect, i.e. to think of the r-proposition $\ulcorner L\ \mathsf{whenever}\ C\urcorner$ as meaning "C is a minimally sufficient cause for F", and the c-proposition $\ulcorner A\ \mathsf{initiates}\ F\ \mathsf{when}\ C\urcorner$ as meaning "C is a minimally sufficient set of conditions for an occurrence of A to have an initiating effect on F".

We include "minimally" here to express our feeling that it's not intuitive to include completely irrelevant fluents in the set C. Hence, as we indicated before, if we were to extend the semantics and entailment relation to include h-, c- and r-propositions, we really would want such propositions to be entailed if and only if they were in the domain description, at least for the simple classes of domain descriptions we've defined so far. (Hence, strictly speaking, we might want to forbid pairs of statements within a single domain description such as $\ulcorner L$ whenever $C1\urcorner$ and $\ulcorner L$ whenever $C2\urcorner$ where $C1$ was a proper subset of $C2$, because the second proposition is redundant).

However, we retain sympathy for your general arguments about the need, ultimately, for theories in classical logic or similar, and for defining entailment in terms of truth functions (as we've effectively done for t-propositions). It is of course debatable whether such theories need to be centered around the notions of global states and state transitions. One's intuitions and preferences about this are probably coloured by one's experience.

Rob and Tony

Q4. **Michael Gelfond** (3.11)

Dear Tony and Rob. I am trying to understand the relationship between your language $\mathcal{E}$ and language $\mathcal{L}$ by Baral, Provetti and myself.

To do that I need some good intuitive understanding of the meaning of statements of $\mathcal{E}$ and I am having some difficulties here. My feeling is that the meaning really depends on what you call "the structure of time". If time is linear then your happens-at corresponds exactly to our occurs-at and your $\ulcorner F$ holds-at $T\urcorner$ to our $\ulcorner f$ at $T\urcorner$. In both cases we have actual occurrences at moments of time (or actual situations as we call them). If time is branching as in your second example in the paper where T corresponds to the sequence of actions then I do not fully understand the meaning of, say, $\ulcorner A$ occurs-at $S0\urcorner$. If it is still a statement of actual occurrence then I think that $\ulcorner A1$ occurs-at $S0\urcorner$ and $\ulcorner A2$ occurs-at $S0\urcorner$ should cause inconsistency. (In the case of linear time we just have concurrent actions).

The meaning of holds-at also seems to change. Instead of actual observations it becomes hypothetical. If I am right then I think this property of the language should be somewhat stressed. If not then some explanation will help.

The goal of $\mathcal{L}$ (as well as of the work by Pinto and Reiter) was to combine situation calculus ontology with actual history of the dynamical system. Since we have both we can combine reasoning about actual occurrences of actions and observations about values of fluents at particular moments of time with hypothetical reasoning of situation calculus useful for planning, counterfactual reasoning, etc.

Can you (and do you want to) use $\mathcal{E}$ for the same purpose?

My other questions are about your logic program. I do not fully understand your definition of initiation point. Do I understand correctly that it should be changed? If so, what happens with the correctness of logic program?

It may be useful to use some semantics of logic program instead of using SLDNF directly. SLDNF can give some results which are correct w.r.t. your specification even though the program is semantically meaningless (Say, its Clark's completion is too weak or inconsistent, or it does not have stable model, etc.) If you prove that the program is semantically correct one will be able to use this result directly even if your program is run on, say, XDB or SLG (which checks for some loops) and not under Prolog.

Finally, more comments on LP4 will help. I find comments like "*Resolve*(A, B, C, D) is true iff [some English description]" extremely useful. Similarly for disjunctive_form, partition, etc.

A4. **The Authors** (5.11)

Hello Michael, thanks for your question (several questions in fact!). Here are replies to each of your points in turn.

You wrote:

> *I am trying to understand the relationship between your language $\mathcal{E}$ and language $\mathcal{L}$ by Baral, Provetti and myself.*

This is indeed an interesting question, and one that we tried to address to some extent in our first (JLP) paper on $\mathcal{E}$ (see Section 3, last three paragraphs).

You wrote:

> *My feeling is that the meaning really depends on what you call "the structure of time". If time is linear then your* happens-at *corresponds exactly to our* occurs-at *and your* $\ulcorner F$ holds-at $T\urcorner$ *to our* $\ulcorner f$ at $T\urcorner$.

Yes, that seems correct.

You wrote:

> *If time is branching as in your second example in the paper where T corresponds to the sequence of actions then I do not fully understand the meaning of, say,* $\ulcorner A$ occurs-at $S0\urcorner$. *If it is still a statement of actual occurrence then I think that* $\ulcorner A1$ occurs-at $S0\urcorner$ *and* $\ulcorner A2$ occurs-at $S0\urcorner$ *should cause inconsistency.*

Yes, the meaning of statements such as $\ulcorner A$ happens-at $S0\urcorner$ would indeed be hard to dissect if put in this type of domain description, hence we've avoided doing so in our examples.

Our intuition about Situation Calculus terms such as $S0$ and *Result*$(A, S0)$ is that they refer to (hypothetical) periods of time *between* (hypothetical) action occurrences. In other words, for all actions A, $S0$ is the period of time immediately before the (hypothetical) occurrence of A, and *Result*$(A, S0)$ is the period of time immediately afterwards.

Now, in order to simulate Situation-Calculus-like hypothetical reasoning in $\mathcal{E}$, we need to refer to the exact points at which actions (hypothetically) occur. Hence we include extra points in our structure of hypothetical time, such as *Start*(*Result*$(A, S0)$) (written *Start*$([A])$ in our syntax), and require that

$$S0 < \mathit{Start}([A]) < [A]$$

We then write $\ulcorner A$ happens-at *Start*$([A])\urcorner$ to assert that there is indeed a hypothetical occurrence of A just before the hypothetical time-point $[A]$ (i.e. *Result*$(A, S0)$. Once we've included the complete set of assertions such as this in the domain description, we can use the same general principles of initiation, termination and persistence (encapsulated in our Definitions 9 and 13 of a model) to reason about what holds in this branching structure of (hypothetical) time.

Like the Situation Calculus and the Language $\mathcal{A}$, with time structures such as this everything is intended to be in hypothetical mode, so that, as you suggest, $\ulcorner F$ holds-at $[A1, A2]\urcorner$ should be read as "F is true in the hypothetical situation $[A1, A2]$".

It is straightforward to extend this approach to partially deal with hypothetical reasoning about concurrent actions, by adapting Chitta Baral's and your ideas. Our structure of time would include sequences of sets of action symbols, e.g. $[C1, C2]$, and, for example, h-propositions of the form $\ulcorner A$ happens-at $Start([C1, C2])\urcorner$ for each A in $C2$.

You wrote:

The goal of $\mathcal{L}$ (as well as of the work by Pinto and Reiter) was to combine situation calculus ontology with actual history of the dynamical system.

Yes. A plug for Miller and Shanahan (JLC 1994) is irresistible here! That work had the same aim (as you point out in your papers), and there is perhaps more similarity between $\mathcal{L}$ and [Miller and Shanahan] than with [Pinto and Reiter]. [Miller and Shanahan] also has the advantage that it deals with concurrent, divisible and overlapping actions.

You wrote:

Since we have both [situation calculus ontology and an actual history] we can combine reasoning about actual occurrences of actions and observations about values of fluents at particular moments of time with hypothetical reasoning of situation calculus useful for planning, counterfactual reasoning, etc.

Can you (and do you want to) use $\mathcal{E}$ for the same purpose?

We haven't thought about this a great deal, although it seems possible that hypothetical and "actual" reasoning (for want of a better term) could be combined in $\mathcal{E}$ by an appropriately rich structure of time. (A simple solution might be to index hypothetical time-points such as $[A1, A2]$ with the actual time-point - typically a natural or real number - from which they were being hypothetically projected, and extend the ordering between all time-points appropriately.)

But (at the risk of re-opening an old and seemingly unstoppable debate), at least for planning our first choice would be to use abduction with a linear time structure rather than deduction with a hypothetical branching time structure. Again, there are some remarks about this in the original (JLP) paper on the Language $\mathcal{E}$.

You wrote:

My other questions are about your logic program. I do not fully understand your definition of initiation point. Do I understand correctly that it should be changed? If so, what happens with the correctness of logic program?

The definition doesn't need to be changed. The reply to Michael Thielscher simply fills in the details that we did not include in the paper. So the proof of correctness of the logic programs is unchanged. Also note that the notions of initiation and termination points are implemented in the logic programs using Proposition 2.

You wrote:

> *It may be useful to use some semantics of logic program instead of using SLDNF directly. SLDNF can give some results which are correct w.r.t. your specification even though the program is semantically meaningless (Say, its Clark's completion is too weak or inconsistent, or it does not have stable model, etc.) If you prove that the program is semantically correct one will be able to use this result directly even if your program is run on, say, XDB or SLG (which checks for some loops) and not under Prolog.*

We agree, and we are in fact working on these lines, as we say in the paper towards the end of Section 5. The point is that the present approach gives us a baseline translation that would be accepted by any semantics of logic programs, at least for those cases (as you say) where the corresponding logic program has a meaning under any semantics. Of course, there is also the debate as to whether every logic program should have a meaning, but this is probably not the place to discuss this issue.

You wrote:

> *Finally, more comments on LP4 will help. I find comments like "Resolve(A, B, C, D) is true iff [some English description]" extremely useful. Similarly for disjunctive_form, partition, etc.*

Yes, sorry. *Resolve* is just a simple implementation of a propositional resolution based prover for positive or negative literals. $Resolve(l1, c, l, t)$ means that we can show that l holds by resolution starting from the clause corresponding to $Whenever(l1, c)$ applied at the time instant t. (The details are really not that important, and in fact *Resolve* can be replaced by any sound propositional theorem prover). It first transforms the "implication" of the r-proposition into normal disjunctive form, using the predicate *DisjunctiveForm*, then the *Partition* predicate picks out the literal l that we are interested in proving, and finally we try to show through the predicate *NothingHoldsIn* that the rest of the disjunction is false, by showing that for each of its literals its negation holds. So, as we say above, it is just a simple and naive implementation of resolution.

Rob and Tony.

Electronic News Journal on Reasoning about Actions and Change 131
Vol. 1: *131-136*, 30.11.1997
http://www.ida.liu.se/ext/etai/actions/nj/9711-1/

Protocol of the discussion during November, 1997 on

Methodology of Research on Reasoning about Actions and Change

Edited by: Erik Sandewall
Linköping University, Sweden

From: Michael Gelfond on 3.11.1997

I found parts of it difficult to follow since I am not sure what the participants mean by the word "theory". Are you referring to theory as organized body of knowledge about some subject matter, to theory in mathematical sense (like theory of probability), or to logical theory - collection of formulae in some language with precisely defined entailment relation? (This is of course a very incomplete list of possibilities).

It is important to be somewhat more precise here because in the AI community "theory" is sometimes identified with an "idea" and I am not sure that it is very useful to publicly judge ideas until they develop into theories. Sometimes this process takes much more than 25 years especially if the idea is prevented from its natural development by premature judgments or if development of a theory requires more than one basic idea.

From: Hector Geffner on 5.11.1997

1. I think the goal in KR/Non-Mon is *modeling*, not logic. A formalism may be interesting from a logical point of view, and yet useless as a modeling language.

   A "solution" is thus a good modeling language:

   declarative, general, *meaningful*, concise, that non-experts can understand and *use*, etc. (I agree with David's remark on teaching the stuff to "cynical" undergrads)

   The analogy to Bayesian Networks and Logic Programs that David makes is very good. We want to develop modeling languages that are like Bayesian networks, but that, on the one hand, are more *qualitative* (assumptions in place of probabilities), and on the other, more expressive (domain constraints, time, first-order extensions, etc).

2. For many years, it was believed that the problem was mathematical (which device to add to FOL to make it non-monotonic). That, however, turned out to be only part of the problem; a part that has actually been *solved*: we have a number of formal devices that yield non-mon behavior (model preference, kappa functions, fixed points,

etc.); the question is how to use them to define good modeling languages

3. The remaning problem, that we can call the *semantic* problem, involves things like the frame problem, causality, etc.

   To a large extent, I think the most basic of these problems have also been *solved*:

   Basically, thanks to Michael and Vladimir, Erik, Ray, and others we know that a rule like:

   if A, then B

   where A is a formula that refers to time i or situation s, and B is a literal that refers to the next time point of situation, is just a constraint on the possible transitions from the the states at i or s, and the following states.

   Or put in another way, *temporal rules are nothing else but a convenient way for specifying a dynamic system (or transition function)*

   Actually, for causal rules, the solution (due to Moises, Judea, and others) is very similar: *causal default rules are just a convenient way for specifying (qualitative) Bayesian Networks*

4. These solutions (that appear in different dresses) are limited (e.g., in neither case B can be an aribtrary formula) but are *meaningful*: not only the work, we can also understand why.

   We also understand now a number of things we didn't understand before.

   e.g., 1. a formula can have different "meanings" according to whether it represents a causal rule, an observation or a domain constraint.

   (this is not surprising from a Bayesian Net or Dynamic systems point of view, but is somewhat surprising from a logical point of view)

   2. reasoning forward (causally or in time) is often but not always sound and/or complete; i.e., in many cases, forward chaining and sleeping dog strategies will be ok, in other cases, they won't.

5. It's not difficult to change the basic solutions to accommodate additional features (e.g., non-deterministic transition functions, unlikely initial conditions, concurrent actions, etc.) in a principled way.

   So, I think, quite a few problems have been solved and default languages, in many cases, are ripe for use by *non* non-mon people.

6. We have to make a better job in packaging the available theory for the outside world, and in delineating the solved problems, the unsolved problems and the non-problems, for the inner community and students.

   Actually I have been doing some of this myself, giving a number of tutorials in the last couple of years at a number of places (I invite you to look at the slides in `http://wwww.ldc.usb.ve/~hector`)

From: Pat Hayes on 5.11.1997

I think this meta discussion, though at times confused (mea culpa, of course), has been useful in revealing a clear divergence between two methodologies, giving different answers to the original question about how we should evaluate work in the field. ("NRAC panel on theory evaluation", ENRAC 21.10).

One view appeals to our human intuitions, one way or another. In this it is reminiscent of linguistics, where the basic data against which a theory is tested are human judgements of grammaticality. We might call this a 'cognitive' approach to theory testing. Talk of 'common sense' is rife in this methodology. Based on the views expressed in these messages, I would place myself, Erik Sandewall, Michael Gelfond in this category. The other, exemplified by the responses of Ray Reiter, Mikhail Soutchanski and Murray Shanahan, emphasises instead the ability of the formalism to produce successful behavior in a robot; let me call this the 'behavioral' approach.

This distinction lies orthogonal to the extent to which people find formality more or less congenial. Both Ray and Erik dislike 'vague claims', and Erik's suggested methodology (Newsletter 23.10) meticulously avoids all contact with psychology, as he emphasises; yet he ultimately appeals to capturing our intuition, rather than any successful application in a robot, to tell us which kinds of model-theoretic structures are more acceptable than others. It also lies orthogonal to the extent to which people see their ultimate goal as that of creating a full-blown artificial intelligence (as both Wolfgang Bibel and Mikhail Soutchanski seem to, for example, along with our founder, John McCarthy), or might be satisfied with something less ambitious. This distinction in approaches - start with insects and work 'up', or start with human common sense and work 'down' - is also a methodological split within AI in general, and seems to be largely independent of whether one feels oneself to be really working towards a kind of ultimate HAL.

Do people find this distinction seriously incomplete or oversimplifying? (Why?) Or on the other hand if they find it useful, which side of the division they would place themselves? In a nutshell, is the immediate goal of the field to understand and accurately model human intuitions about actions, or is it to help produce artifacts which behave in useful or plausible ways? I think this is worth getting clear not to see which 'side' wins, but to acknowledge that this difference is real, and likely to produce divergent pressures on research.

From: Erik Sandewall on 6.11.1997

Pat,

I agree with you that it's time to sort out the different perspectives, goals, and methods for reaching the goals that have confronted each other here. You write of two dimensions; in the first one you make the following distinction:

> *One view appeals to our human intuitions, one way or another. In this it is reminiscent of linguistics, where the basic data against which a theory is tested are human judgements of grammaticality. We might call this a 'cognitive' approach to theory testing. Talk of 'common sense' is rife in this methodology. Based on the views expressed in these messages, I would place myself, Erik Sandewall, Michael Gelfond in this category. The other, exemplified by the responses of Ray Reiter, Mikhail Soutchanski and Murray Shanahan, emphasises instead the ability of the formalism to produce successful behavior in a robot; let me call this the 'behavioral' approach.*

I agree with this, except that the term 'behavioral' is maybe not the best one, and also you put me in the wrong category; more about that later. Anyway, the distinction you make here seems to coincide with the one that David Poole made in his position statement:

There are two quite different goals people have in building KR system; — These are:

1. A knowledge representation as a modelling language. If you have a domain in your head you can use the KR to represent that domain. —

2. A knowledge representation as a repository of facts for commonsense reasoning. Under this scenario, you assume you are given a knowledge base and you are to make as much sense out of it as possible. —

If you are going to *design* a robot in a good engineering sense, you are going to need to model both the robot itself and its environment. That's why what you call the 'behavioral' approach coincides with the use of KR for modelling physical systems. Since 'modelling' can mean many things, I'll further qualify it with the term 'design goal'.

As for the other dimension, you propose

— the extent to which people find formality more or less congenial. Both Ray and Erik dislike 'vague claims' —

This distinction I find less informative, since all the work in this area is formal in one way or another. Even the kludgiest of programs exhibits 'formality'. However, different researchers do take different stands wrt how we choose and motivate our theories. One approach is what you described in your first response to the panel (ENRAC Newsletter on 22.10):

Knowledge-hackers try to formalise an intuition using logic A and find it hard to match formal inference against intuition no matter how ingenious they are with their ontologies and axioms; so they turn to logic B, which enables them to hack the examples to fit intuition rather better.

The key word here is *examples*. In this example-based methodology, proposed logics are treated like hypotheses in a pure empirical paradigm: they are accepted until a counterexample is found; then one has to find another logic that deals correctly at least with that example. Ernie Davis characterized this approach in his book, Representation of Commonsense Knowledge [mb-Davis-90]. See also the discussion of this approach in my book, Features and Fluents [mb-Sandewall-94], p. 63).

The example-based methodology has several problems:

- It does not prove anything, just like you can not prove the correctness of a program by test examples, and particularly not by trying it on a small number of toy tests.
- The process does not seem to converge; it does not even allow us to believe in any one hypothesis for a very long time. Here's why the good old positivist methodology doesn't work in this case: the empirical data are notoriously unreliable. We may be reasonably clear

> about what are the commonsense conclusions for a simple scenario that is presented to us, but when more complex and sophisticated scenarios are admitted, we get more and more cases where common sense is not held in common. Therefore, people will *always* come up with proposed new counterexamples to any given theory.

The choice of methodology is indeed orthogonal to your first distinction, since the example-based methodology can be used both in the pursuit of theories of common sense, and in the development of intelligent robots by design iteration (try a design, see how it works, revise the design).

The alternative to this is to use a systematic methodology where, instead of searching for the "right" theory of actions and change, we identify a few plausible theories and investigate their properties. For this, we need to use an underlying semantics and a taxonomy of scenario descriptions; we can then proceed to analyse the *range of applicability* of proposed theories (entailment methods).

Your answer to this was (31.10):

> *Yes, but to what end? The things you characterize as 'ill-defined' are the very subject-matter which defines our field. There is no objective account of 'action', 'state', etc. to be found in physics, or indeed in any other science; intuitions about these things is the only ultimate test we have for the correctness or appropriateness of our formalisms.—*

This would be true if the 'cognitive' (in your terms) goal were the only one. From the point of view of modelling and design, on the other hand, these are perfectly valid concepts. The concept of state is used extensively in control engineering (yes, control theory does deal with discrete states, not only with differential equations!), and I am sure our colleagues in that area would be most surprised to hear that our intuitions is "the only ultimate test we have" for the correctness or appropriateness of the formalisms that they share with us.

Now, when you placed me in the cognitive category, you got me wrong. As I wrote in my position statement for this panel, my heart is with the use of knowledge representations as modelling languages. The present major project in our group is concerned with intelligent UAV:s (unmanned aircraft), and in this enterprise we need a lot of modelling for design purposes; we have currently no plans to pursue the 'cognitive' goal.

However, just as the example-driven methodology can serve both the cognitive goal and the design goal, I do believe that the systematic methodology can *also* be relevant as one part of a strategy to achieve the 'cognitive' goal. More precisely, for the reasons that both you and I have expressed, it's not easy to find any credible methodology for research on understanding the principles of commonsense, and in fact I did not see any concrete proposal *for* such a methodology in your contributions. However, to the extent that people continue to pursue that goal, my suggestion was to divide the problem into two parts: one where our discipline can say something substantial, and one which is clearly in the domain of the psychologists.

Therefore, the contradiction that you believed having seen when writing

> *... and Erik's suggested methodology (Newsletter 23.10) meticulously avoids all contact with psychology, as he emphasises; yet he ultimately*

> *appeals to capturing our intuition, rather than any successful application in a robot, to tell us which kinds of model-theoretic structures are more acceptable than others.*

is not a real one; it only arises because your perception that

> *... this distinction in approaches - start with insects and work 'up', or start with human common sense and work 'down' - is also a methodological split within AI in general, and seems to be largely independent of whether one feels oneself to be really working towards a kind of ultimate HAL.*

which I also do not share. After all, the behavioral/ commonsense view and the modelling/ design view represent *goals*, not methodologies, and both choices of methodology (the example-based and the systematic one) can be applied towards both the goals.

References:

[mb-Davis-90] Ernest Davis. *Representation of Commonsense Knowledge.* Morgan Kaufmann Publishers, Inc., 1990.

[mb-Sandewall-94] Erik Sandewall. *Features and Fluents. The Representation of Knowledge about Dynamical Systems.* Oxford University Press, 1994.

Electronic News Journal on Reasoning about Actions and Change 137
Vol. 1: *137-155*, 30.11.1997
http://www.ida.liu.se/ext/etai/actions/nj/9711-2/

Protocol of the discussion during November, 1997 on

Ontologies of Actions and Change

Edited by: Erik Sandewall
Linköping University, Sweden

From: Michael Gelfond on 3.11.1997

I agree with Ray that it may be a good idea to separate "ontology" from "epistemology".

For me to specify "ontology" means to give a collection of individual objects, functions and relations which comprise our domain. The type of logical connectives used to construct sentences in this language is part of "logical system" and does not belong to the ontology. Closed World Assumptions of different sorts,etc belong to "epistemology".

We seem to differ slightly on the meaning of the frame problem. To me the frame problem is a special case of a more general problem of finding logical system suitable for representing defaults and their exceptions in a way which will insure a high degree of elaboration tolerance. (Frame axiom is a particularly difficult default since it is related to causality, representation of time, etc)

This is of course only one of several possible views but I think an important one.

I believe that the ontology of situation calculus and action languages are basically the same. Situation IS a sequence of actions, etc. Our causal models (or automata) are graphs representing trees of situations. (One superficial difference is that our actions are undefined in situation where the corresponding preconditions are not satisfied, while in Ray's case they seem to "return" the same situation).

The more important difference is that action languages stress the distinction between description languages and query languages and define entailment relation directly without appealing to first-order logic.

To turn these languages into calculi one need to decide what logical system to use to formalize reasoning about causal models defined by the corresponding action theory. My favorite choice here is declarative logic programming with answer set semantics which can be viewed as a variant of default logic.

We have now several logic programming versions of situation calculus proven to be sound and complete w.r.t. entailment in action description languages.

The disadvantage of this approach is that we need to develop more mathematics for dealing with declarative logic programs (default theories) which is a slow process. Even though there has been an important progress in this in the last few years, classical logic is still a better choice in this respect. I believe however that in a long run default logics will allow for more elaboration tolerant and computationally better representations of various types of commonsense knowledge.

This is related also to the comment by Mikhail Soutchanski when he stresses the differences between classical and non-classical logics for knowledge representation. If formalizations of different domains are all done in default logic then we will have no problem to combine all of them together. For instance, if our initial situation is defined in Datalog then it combines very nicely with logic programming version of situation calculus. Even if the languages used for encoding knowledge in different domains are special purpose (like action languages) we will not have any problem combining them together if their entailment relations are formalized in the same language. (e.g. Instead of $\mathcal{A}$ we can use its sound and complete formalization in default theories).

I am not claiming BTW that nonmonotonic logics are better than classical one. I do not take it as a truth (revealed or otherwise). Pat (and Ray?) can be right and commonsense reasoning can be compatible with monotonic logic. But I think that it is an open and difficult research question which can be answered but trying various approaches.

One more comment on how action languages are used. Suppose you have some collection of actions and their effects and would like to write a planner which examines, in some reasonable order, possible sequences of actions and checks if your domain description entails that the goal G holds after the sequence of actions is executed. In this case domain description in $\mathcal{A}$-like language together with its entailment relation can be used to precisely specify the problem.

The corresponding logic programming formalization of this entailment together with the domain description forms the main part of the planner. If you decide to use Prolog inference engine you may need to slightly transform the program to avoid floundering, etc. If you do it using results from the theory of logic programming the resulting program will be provenly correct. It seems to be very similar to what people in Toronto do, except we probably concentrate more on correctness of actual program.

There are of course many other uses.

Finally, some corrections to Ray's comment on $\mathcal{A}$-languages. They have histories (actually, it is their basic feature). Language $\mathcal{L}$ from this family combines branching and linear time. This is published in [j-jlp-31-201], which is a special issue on Logic Programming and Reasoning about actions and I think it contains good papers.

As Vladimir already mentioned there are (fully) first order languages.

References:

[j-jlp-31-201] Chitta Baral, Michael Gelfond, and Alessandro Provetti. *Representing Action: Laws, Obervations and Hypotheses.* Journal of Logic Programming, vol. 31 (1997), pp. 201-244.

From: Mikhail Soutchanski on 3.11.1997

I will follow the distinction between the gof-sitcalc (sitcalc of 1969) and the R-sitcalc (sitcalc of 1997+); this distinction is proposed by Pat Hayes in

ENRAC 31.10.1997.

> The situation calculus is a foundation for general purpose high-level programming languages. Note that it is an easy exercise to formalize the Turing machine in the SC.

It is pretty easy to formalize a Turing machine in almost anything, so this doesnt mean much. But in any case, what has this got to do with the topic we are all concerned with? How did Turing machines come to be relevant here?

My intention was to turn the attention to computational aspects: the R-sitcalc is a general theory of action that is a foundation for the high-level logic-programming language GOLOG and its descendants (mentioned by Ray Reiter, ENRAC 27.10.1997). Because a version of GOLOG is used to control a robot in the real world, the R-sitcalc (as a formalism for reasoning about actions) can be judged according to the test suggested by Murray Shanahan (ENRAC 23.10.1997).

> Most intuitive reasoning done by humans lies entirely outside the purview of the situation calculus.
>
> Note that your objection can be easily rephrased as: "Most intuitive reasoning done by humans lies entirely outside the purview of the formal logic".

I have no idea what this response is supposed to mean. Do you identify formal logic with the situation calculus? Or do you mean only that much of intuitive reasoning is outside the scope of our subject? Or what??

I do not identify formal logic with the situation calculus. But it is true that I am not sure whether formal logic corresponds precisely to much of our intuitive reasoning. My point was that it is not fair to judge the R-sitcalc only according to the criterion whether it exactly captures *our* intuitions about actions, changes and situations (by the way, nobody ever claimed that the gof-sitcalc or the R-sitcalc expresses the final truth). As long as robots can successfully execute high-level programs based on the R-sitcalc, it is not completely worthless.

> I'm not sure whether we must have the same concerns that the cognitive science has. Most of the people do not think in terms of C, LISP, PROLOG, but all these languages are still useful for writing programs that will exhibit an intended behavior. Similarly, the SC is useful as the basis for the high-level programming language.

And again, I'm at a loss to understand your point. What is 'the high-level programming language' you refer to here? For the record, I think that unless our subject is part of cognitive science, then it's just empty formula-hacking. See earlier comments in reply to Erik.

'The high-level programming language' = GOLOG. There are several experiments performed by anthropologists and psychologists with different people in different parts of the world. As far as I understand their results, in some cases, ability to solve naive physics problems or derive conclusions from syllogistic premises depends on cultural background, education and other personal factors. For this reason, psychology (and cognitive science) cannot be *the only* foundation of our research.

From: Rob Miller on 6.11.1997

Hector,

I'd like to express agreement with your first point, that KR is about modelling. But I'd like to take issue with a couple of your other points, (3) and (5). In point (3) you said:

> *The remaning problem, that we can call the* semantic *problem, involves things like the frame problem, causality, etc.*
>
> *To a large extent, I think the most basic of these problems have also been* solved*:*
>
> *Basically, thanks to Michael and Vladimir, Erik, Ray, and others we know that a rule like:*
>
> *if A, then B*
>
> *where A is a formula that refers to time i or situation s, and B is a literal that refers to the next time point of situation, is just a constraint on the possible transitions from the the states at i or s, and the following states.*
>
> *Or put in another way,* **temporal rules are nothing else but a convenient way for specifying a dynamic system (or transition function)**
>
> ......

My problem with this is that, in general, dynamical systems in the everyday world can't be realistically modelled as state transition systems, because they involve things like continuous change, actions or events with duration, partially overlapping events, interuptable events, etc. That's why other communities involved in modelling dynamical systems (e.g. physicists, engineers, the Q.R. community) choose to model time as the real numbers. In this case, there is no "next time point", so it's difficult to read "if A, then B" as a constraint in the way you suggest. The analogy between everyday dynamical systems and state transition systems/database updates only works for a relatively small class of carefully picked domains.

Your point (5) was:

> *It's not difficult to change the basic solutions to accommodate additional features (e.g., non-deterministic transition functions, unlikely initial conditions, concurrent actions, etc.) in a principled way.*

Well again, it seems to me that if this is true, it's simply because researchers tend to pick "additional features" to work on which will conveniently fit into the state transition view of the world, as opposed to picking from the rather large collection of issues that won't.

From: Ernie Davis on 6.11.1997

Ray Reiter writes, in newsletter ENRAC 23.10

> *qualitative physics and planning have no difficulty with the FP because, without exception, they adopt the STRIPS sleeping dogs strategy. Which is to say,* **they assume they have complete information about world states**.

I don't think that this is quite right in the case of qualitative physics. My KR '92 article "Axiomatizing Qualitative Physics" [c-kr-92-177] presents a theory which, being in first-order logic, is perfectly able to characterize inferences from partial information, but does not require any special frame axioms for the continuous parameters. The reason is that the behavior of a continuous parameter is governed by governed by a qualitative differential equation of the form, "The derivative of P is the sum of the influences on P". P remains absolutely constant if the sum of the influences is zero. P retains the same qualitative value to some next modes of the system if it is consistent that some other parameter should change its value before P does. In any case, the behavior of P in staying the same is governed by the same law that governs its behavior in changing. No special law is needed to cover the cases where P stays the same. (For discrete parameters, I did need a frame axiom.)

More generally, for those, like Pat and me, whose primary interest is physical reasoning, a temporal ontology whose central category is "a finite sequence of actions" seems awkward at best. Physical reasoning is chiefly concerned with continuous, asynchronous, external change, and it is much easier to deal with this by making the continuous time-line primary and adding actions on top of that, rather than vice versa.

– Ernie Davis

References:

[c-kr-92-177] Ernest Davis. *Axiomatizing Qualitative Process Theory.* Proc. International Conf on Knowledge Representation and Reasoning, 1992, pp. 177-188.

From: Hector Geffner on 10.11.1997

Rob Miller says:

> *My problem with this is that, in general, dynamical systems in the everyday world can't be realistically modelled as state transition systems, because they involve things like continuous change, actions or events with duration, partially overlapping events, interuptable events, etc. ...*

My point is that action languages - in any dress you like - are just a convenient means for specifying (and in certain cases reasoning with) dynamic systems. That is the main lesson I think of the Yale Shooting Problem(s) and a lot of the work on temporal non-mon. Namely, the meaning of a rule like:

```
if loaded and shoot then not alive
```

is that the only state trajectories s0, s1, s2, ..., that are possible are the ones in which 'alive' is false at si+1 when loaded and shoot are true at si.

You can formulate the idea in many ways (suitable circumscriptive policy, Erik's version of chronological minimization, predicate completion, ...), but it is the same idea: *rules specify possible state transitions, observations prune possible state trajectories.*

Now, Rob is right; dynamic systems come in different varieties; e.g.,

1. discrete time, discrete value space
2. discrete time, continuous value space
3. continuous time, continuous value space
4. ....

Rules like the one above (with first order extensions, etc) are good for specifying systems of Type 1 only. Yet it's not difficult to see how systems of Type 2 could be specified as well.

Actually there are *other* type of mathematical models for the type of problems that Rob has in mind as the "Semi-Markov Decision Processes" (probabilistic continuous processes - like queuing systems - that are controlled at discrete time intervals).

My point is that we are *not* inventing new mathematical models of dynamic systems. What we are inventing are suitable structured languages for specifying and in certain cases controlling those systems. That's what STRIPS is about.

In my view, the KR/control enterprise is about developing richer versions of STRIPS suitable for specifying and controlling not only systems of Type 1, but also Markov Decision Processes, Partially Observable MDPs, Semi-MDP's, etc etc.

How we will measure success?

When we can model and control some dynamic systems that cannot even be modeled using non KR methods.

- Hector Geffner

From: Rob Miller on 11.11.1997

Hector Geffner (ENRAC 10.11) wrote:

> *Now, Rob is right; dynamic systems come in different varieties; e.g.,*
>
> 1. *discrete time, discrete value space*
> 2. *discrete time, continuous value space*
> 3. *continuous time, continuous value space*
> 4. *....*
>
> *Rules like the one above (with first order extensions, etc) are good for specifying systems of Type 1 only. Yet it's not difficult to see how systems of Type 2 could be specified as well.*
>
> *Actually there are* other *type of mathematical models for the type of problems that Rob has in mind as the "Semi-Markov Decision Processes" (probabilistic continuous processes - like queuing systems - that are controlled at discrete time intervals).*

That's right. But I think that an important wider problem that we have to tackle within "reasoning about action and change" is how to synthesise or combine very different approaches to modelling dynamic systems within a single "commonsense" framework. For example, I'd like to see more research along the lines of Erik Sandewall's 1989 work on combining reasoning about actions with modelling using the differential calculus. It's true that there has been a small amount of subsequent work on this theme since then (see e.g. `http://www.dcs.qmw.ac.uk/~rsm/project.html#Other` for a bibliography). But not much compared with, say, work on extending state-transition based approaches to deal with ramifications in evermore sophisticated ways. Why is this so? Why don't we put much effort into addressing challenges such as Kuipers' - on combining the Situation Calculus with Q.R. (see Kuipers' book, p. 201)? If we did more of this type of work, we'd stand more chance of being able to (in Hector's words) "package the theory for the outside world".

Rob

From: Erik Sandewall on 13.11.1997

Rob,

You wrote:

> *... But I think that an important wider problem that we have to tackle within "reasoning about action and change" is how to synthesise or combine very different approaches to modelling dynamic systems within a single "commonsense" framework. For example, I'd like to see more research along the lines of Erik Sandewall's 1989 work on combining reasoning about actions with modelling using the differential calculus. It's true that there has been a small amount of subsequent work on this theme since then ... But not much compared with, say, work on extending state-transition based approaches to deal with ramifications in evermore sophisticated ways. Why is this so? Why don't we put much effort into addressing challenges such as Kuipers' - on combining the Situation Calculus with Q.R. (see Kuipers' book, p. 201)? ...*

Unfortunately, the answer to this question is a brutal one: publication problems. At least, that's the conclusion I have drawn from the experience of our group. The following is what happened after our start on hybrid systems in 1988-89. A key new result in the 1989 papers [c-kr-89-412], [c-ijcai-89-894] was that minimization or restriction of change generalized nicely to minimization or restriction of discontinuities. (The particular use of chronological minimization as a restrictor on the set of models was of secondary importance, I think; one can do it in other ways). The weak spot that we identified at the same time, and which was clearly spelled out in the papers, was that some additional model selection criterion was necessary, since we still got some unintended models. There were two options: modifying the logic itself, or introducing concepts from other disciplines.

The first approach was pursued by two of our graduate students at the time, Tommy Persson and Lennart Staflin. Their first paper in this direction was accepted at ECAI 1990 [c-ecai-90-497], but then they ran into the wall. One more of their papers is still available as a departmental report; it was called "Cause as an Operation on a Logic with Real-valued Fluents and Continuous Time". The article is available at the URL

`www.ida.liu.se/publications/cgi-bin/tr-fetch.pl?r-90-45+abstr`

and the abstract goes as follows:

> *We propose a new method for characterizing the discontinuities in processes that are mostly continuous. We introduce a causal operator*

that is used to specify when the value of a fluent has a cause. A discontinuity in a fluent is allowed if the fluent's value immediately after the discontinuity has a cause. The causal operator is incorporated in a temporal logic with continuous time and real-valued fluents. The resulting logic is a nonmonotonic logic suitable for representing physical models of real world situations. We define a selection function which given a set of models returns a subset of the models. This selection function defines a nonmonotonic entailment operator. The intuitive idea behind the selection function is that is should select all models where all discontinuities are "specified" as allowed.

In other words, they proposed what is known today as a causal approach. The paper was rejected for IJCAI 1991. Around the same time, my journal style article which combined and extended the results in the 1989 KR and IJCAI papers was rejected for the A I Journal, with vitriolic reviews.

The other approach, which we also investigated, was to bring in aspects of real physics. We started cooperation with people who had that competence, and in particular with our colleagues in control theory. This led to work on the use of bond graphs, which is a classical energy-based method for modelling physical systems, and uniformly applicable to systems from different domains (mechanical, electrical, hydraulic, etc.). Members of our group (Strömberg, Söderman) developed a generalization of bond graphs to take account of abrupt changes (that is, combining continuous and discontinuous) by introducing a "switch" concept in a clean way.

Yet another approach was the use of hybrid transition systems, which are a generalization of the transition systems that come from the theory of real-time systems. Additional members of our group (Nadjm-Tehrani and Strömberg) used hybrid transition systems for modelling actions, analyzing safety conditions ("is it possible that if I drive this way, I may crash into the car in front of me?"), etc.

For both bond graphs and transition systems, the idea was to import methods from other areas into AI and KR. In both cases, our people were able to publish successfully in the neighboring discipline, *but not in AI*, or at least most of the AI submissions were rejected. Reviewers tended to say either that this was not relevant for AI, or that although possibly relevant, more would have to be done in order to reach the presumed high quality standards that we require in our field.

It goes without saying that after a few experiences of this kind, these (then) Ph.D. students turned away from AI and continued their work in the areas where they were better received. They were also put off by what they considered as idiotic comments by reviewers, for example, to the effect that the proposed modelling system was not capable of accounting for the sudden occurrence of asteroids on the scene.

When these things happen, it is our discipline that stands to lose. There were great opportunities at that time for bringing in fresh concepts into KR, and for integrating them with what we are otherwise doing. On the other hand, time does not stand still while we fumble, and if our area does not deal in a timely fashion with new problems, then there are others who will.

It is also important to note that this resistance to new ideas is not reciprocal. This year's HART conference (Hybrid And Real Time systems) had no trouble accepting my paper on relating high-level and low-level descriptions of actions, which was an extension of my invited paper at last year's ECAI.

This panel discussion has already touched on the remarkable persistence of situation calculus in our field. The field's unwillingness to accept and use outside knowledge for dealing with continuous change is equally remarkable.

Erik

References:

[c-ecai-90-497] Tommy Persson and Lennart Staflin. *A Causation Theory for a Logic of Continuous Change.* Proc. European Conference on Artificial Intelligence, 1990, pp. 497-502.

[c-ijcai-89-894] Erik Sandewall. *Filter Preferential Entailment for the Logic of Action in Almost Continuous Worlds.* Proc. International Joint Conference on Artificial Intelligence, 1989, pp. 894-899.

[c-kr-89-412] Erik Sandewall. *Combining Logic and Differential Equations for Describing Real-World Systems.* Proc. International Conf on Knowledge Representation and Reasoning, 1989, pp. 412-420.

From: Patrick Doherty on 17.11.1997

After following the discussion between Rob Miller and Tom Costello, I'd like to point out another approach that my group has been using in our research in the area of action and change. It is based on a distinction between surface and base languages made by Sandewall in Features and Fluents. The family of logics we use is called TAL (Temporal Action Logics) and the newer versions are generalizations of an entailment policy called PMON, first described in F&F.

Clearly, high-level narrative description languages are not only useful, but will obviously be necessary when dealing with scenarios more complex than those we see in the literature today. On the other hand, general purpose logics such as classical logic have great advantage when doing comparative analyses, debugging and incremental extension of formalisms.

In our approach, we combine the advantages of each. Our narrative descriptions are represented in terms of a high-level language which allows for straightforward description of observations, action instances and types, casual rules, and explicit temporal constraints. The high-level language may simply be viewed as a set of macros where each has a modular translation into formulas in the base language, 1st-order classical logic. The language is always extensible in an incremental manner. So far, we've extended the language for causal rules and concurrency simply by adding new macros and translation functions.

The logic TAL 1.0, and the approach using surface and base languages is implemented and accessible as an applet or a Marimba Castanet Channel. The visualization tool allows for the construction of narratives in the high-level language, their automatic translation into a 2nd-order theory, and that theories automatic translation into a first-order theory. One also has the possibility of viewing models as time-lines and a query mechanism is provided. The system and related references are accessible via the following URL:

`http://anton.ida.liu.se/vital/vital.html`

The majority of scenarios discussed in the literature are represented in the tool and can be queried. The purpose of the tool is not only for our individual research, but also to open up the logics for public evaluation and comparative analyses. The use of both a high-level macro language and a translation into classical logic should meet the needs of groups taking the $\mathcal{A}$ language approach or those more comfortable with good old classical logic.

The danger we find with the trend in using $\mathcal{A}$ language approaches is that it often appears to be the case that one is taking a relatively simple surface language and translating into what turns out to be something along the lines of classical logic, but in a rather indirect and complex manner. It is difficult to see how the guarantee of semantic continuity in the base language or incrementality in the surface language will be met as scenarios or narratives become increasingly more complex. On the other hand, if provided with well-understood and modular translations into classical logic, it is much easier to evaluate progress and simplify comparisons. One sign that there is a problem is that the $\mathcal{A}$-type languages are generally only compared relative to other $\mathcal{A}$-type languages. Of course, translations of formalisms to classical logic and ensuing comparisons are not all that simple when comparing widely differing ontologies, but we have a rich infrastructure of well-established technical tools to help us along.

I'm certainly all for the flourishing of alternative approaches to modeling action and change, but I really think it is time to clean up our methodology, do more comparative analyses across paradigms regarding strengths, weaknesses, assessments of use, and to apply the formalisms to some "real" problems in the area of DES, process control, etc. I'd like to see a library of tools and implementations which allow each of the different groups to actually test the representational capabilities of the perspective approaches and a number of realistic modeling challenges similar to those one finds in the "Hybrid Systems" area. This appears to be a common and useful aspect of methodology in other areas. Why is this lacking in our area and what can we do about it?

I hope the tool we have developed and placed on-line might serve as a starting point for discussion or for developing healthier methodological tools and coherence in the area. Perhaps Murray Shannahan's and Ray Reiter's interests in controlling robots with logics could also lead to another set of testbed's for comparative analysis of formalisms.

From: Vladimir Lifschitz on 19.11.1997

I would like to respond to some of the comments on action languages published in ENRAC 13.11 and 17.11.

Tom Costello writes to Tony Kakas and Rob Miller regarding their new action language:

> *The reason I ask for truth conditions for your propositions is that I cannot understand what the intuitive consequences of a set of propositions should be, unless I understand what the propositions say.*
>
> *As you say, action languages are supposed to be "understandable and intuitive". Languages cannot be understood without semantics.*

It seems to me that the semantics of an action language cannot be described by specifying truth conditions for its propositions. The problem is the same as with nonmonotonic languages in general. Take, for instance, the closed-world database P(1),P(2). The negation of P(3) is a consequence of this database, but this fact cannot be justified on the basis of truth conditions for P(1) and P(2).

Patrick Doherty writes:

> *The danger we find with the trend in using $\mathcal{A}$ language approaches is that it often appears to be the case that one is taking a relatively simple surface language and translating into what turns out to be something along the lines of classical logic, but in a rather indirect and complex manner.*
>
> *On the other hand, if provided with well-understood and modular translations into classical logic, it is much easier to evaluate progress and simplify comparisons.*

It is impossible, unfortunately, to translate an action language into classical logic in a modular way, because classical logic is monotonic, and action languages are not. The best we can achieve is a modular translation from an action language into a nonmonotonic formalism, such as circumscription, whose semantics can be characterized by a nonmodular translation into classical logic. This is indeed indirect and complex. But we have to pay this price for the convenience of reasoning about actions in classical logic.

Vladimir Lifschitz

From: Erik Sandewall on 21.11.1997

Vladimir,

In ENRAC 19.11, in the context of the discussion with Tom Costello and Patrick Doherty, you wrote:

> *It is impossible, unfortunately, to translate an action language into classical logic in a modular way, because classical logic is monotonic, and action languages are not. The best we can achieve is a modular translation from an action language into a nonmonotonic formalism, such as circumscription, whose semantics can be characterized by a nonmodular translation into classical logic. This is indeed indirect and complex. But we have to pay this price for the convenience of reasoning about actions in classical logic.*

Through the Features and Fluents approach, we have demonstrated a much more direct and less complex way of doing these things. However, "convenience of reasoning" is not the only issue, and it's not what Patrick addressed; what he actually wrote and what you quoted in the preceding lines was:

> *On the other hand, if provided with well-understood and modular translations into classical logic, it is much easier to evaluate progress and simplify comparisons.*

In particular, translating scenario descriptions into first-order logic helps evaluation and comparisons in two ways. First, for transparency, i.e. for allowing us to understand in a precise manner what the scenario descriptions say, which is also what Tom Costello's persistent questions are concerned about. After it, there is the issue of actually carrying out the reasoning about actions, which quite possibly can be done (or implemented) more conveniently if one uses first-order theorem provers as inference engines. Anyway, let's stick to the question of how we can evaluate progress and simplify comparisons for the work that gets done and published.

With respect to transparency, it seems to me that action languages such as $\mathcal{A}$ and $\mathcal{E}$ add to the obscurity rather than dissolving it. In the Features and Fluents approach, we achieve the same results in a much less elaborate fashion: we have *one single language*, namely a straightforward multi-sorted first-order theory; we have a set of *syntactic abbreviations* or "macros" in order to sugar the notation for legibility; and we have *two different semantics*. There is the classical semantics of the Tarskian type which is almost straight from the textbook, with routine modifications to take care of the multitude of types and for assuring a closed-world assumption with respect to objects. There is also the *underlying semantics* which specifies what one really means - corresponding to Tom Costello's question to Tony Kakas and Rob Miller, where he wrote:

> *The reason I ask for truth conditions for your propositions is that I cannot understand what the intuitive consequences of a set of propositions should be, unless I understand what the propositions say.*
>
> *As you say, action languages are supposed to be "understandable and intuitive". Languages cannot be understood without semantics.*

The underlying semantics does exactly this. To put it another way, here is a recipe for converting an action-language approach to our approach. You take the action-language setup with its two languages, each with its own syntax and semantics. First, you remove the separate syntax of the action language. You retain the use of two distinct semantics, so one and the same scenario description can be mapped into a set of models by two different methods: the set of classical models, and the set of intended models. Also, you make sure that the two semantics use the same space of possible interpretations; it's just that the set of intended models is (usually) a subset of the set of classical models. The you have captured the essentials of our approach.

The definition of the underlying semantics is indeed not truth-functional in a conventional way; it can rather be described as a kind of simulation of the world in question. However, truth conditions are still used within that 'simulation', namely for defining the truth conditions for each individual observation statement. The resulting semantics is concise, intuitively convincing, and formally precise at the same time.

This approach has several important advantages:

- No need to define new languages all the time, and of comparing newly published languages with previously published ones. We stay with the same language, which is sufficiently expressive right from the start to last for a while. In particular, it uses an explicit time domain and allows both non-metric "successor" time, integer time, and real time. The time domain may be either linear or forward-branching. Multi-valued fluents are allowed; objects are allowed so the language is "first-order" rather than "propositional", and others more.

- New problems can be addressed with very small initial effort.

Consider ramification, for example. In the $\mathcal{A}$ tradition, new language variants were introduced for ramification. In our approach, all you have to do is to remove the syntactic restriction that is imposed in the footnote on page 176 in the "Features and Fluents" book, select PMON among the twelve entailment methods that are defined and analysed there, and you have a method for ramification that in fact is as powerful or more powerful than most of what has been published in the last couple of years. (To be precise, the two formulations of PMON that are stated on page 243 are no longer equivalent after the generalization, and you have to use the second one. That's all).

- Consequently, you can get much more quickly to the point where you actually *analyse* the entailment methods in order to verify their range of applicability. You don't spend so much of your time setting up the definitions.

A possible objection against defining a quite expressive language from the start is that you may not be able to prove your results for such a broad language. That is one reason why we focus on "range of applicability" results rather than "validation" results. Instead of defining a narrow language and proving that a property holds throughout the language, we identify what is the part of our broad language where the property holds. This helps avoiding the proliferation of languages, but it also helps obtaining results that are as general as possible, since the result is not artificially constrained by the choice of language. This is a real difference between the approaches, since the published general results about $\mathcal{A}$ type languages are consistently validation results (unless I have missed something).

Then there is the other reason for reducing an action language to a first-order theory, namely for implementation purposes. There, again, Doherty et al have developed efficient computational mechanisms as a part of our approach; their implementation has been available for on-line use over the net since May of this year. In their case it's only an implementation issue; the purpose of their work is not to assign a meaning to the language since that has already been taken care of. (Patrick writes more about this in his contribution to the discussion, below).

The bottom line is that it is perfectly possible to achieve legibility (initial motivation of action languages), transparency (Tom Costello's request), and effective implementation by the approach used in Features and Fluents, and with much less administrative overhead than in the work based on action languages. My question is, therefore: what are the real benefits of all the definitional work in the action-language papers; what are the results that the rest of us can use?

From: Tom Costello on 21.11.1997

Vladimir writes,

> *It seems to me that the semantics of an action language cannot be described by specifying truth conditions for its propositions. The problem is the same as with nonmonotonic languages in general. Take, for instance, the closed-world database P(1),P(2). The negation of P(3) is a consequence of this database, but this fact cannot be justified on the basis of truth conditions for P(1) and P(2).*

I do not ask that Action language designer's define their semantics in terms of truth conditions. I ask that the designers give truth conditions to each of their assertions. The difference can be seen in your example if you take P to mean there is a flight and 1 to means Glasgow,London and 2 to mean London,Moscow. The P(1) is true, if there is a flight from Glasgow to London. What is puzzling me about Kakas and Miller's language is what their propositions mean, in exactly this sense.

In particular, what does

```
A causes F if G
```

or

```
A initiates F if G
```

mean. That is, given a model M of a domain description, when is this proposition satisfied in M. I have pointed out that problems arise under certain definitions of model.

The most difficult issue that I am aware of, is that it is unclear whether

```
A causes F if G
```

means that

```
in every actual state S where G is true,
    then F is true in R(A,s),
```

or the similar, but different

```
in every possible state S where G is true,
    then F is true in R(A,s).
```

Similarly, does

```
Always F,G
```

or Kakas and Miller's

```
F Whenever -G
```

mean that every actual state satisfies F,G, or every possible state.

Tom Costello

From: Patrick Doherty on 21.11.1997

Vladimir's reply to my paragraph about $\mathcal{A}$ Languages introduces two interesting and subtle issues:

1. What is meant by modular/non-modular translation.
2. What is meant by a monotonic/nonmonotonic approach.

I am not sure if these topics belong to the ontology panel, but they are highly relevant when dealing with comparative analyses across formalisms and understanding what on the surface appear to be opposing "ideological stances" in methodology.

To clarify my "current" working view of the role action languages should play, I simply quote the original intuition Vladimir himself at one time had about their role in his paper, "Two Components of an Action Language":

Originally, action languages were meant to play an auxiliary role. The primary goal was to represent properties of actions in less specialized formalisms, such as first-order logic and its nonmonotonic extensions, and the idea was to present methods for doing that as translations from action languages.

My group currently uses this approach and it is also one of the cornerstones of the Features and Fluents methodology. Formally and conceptually, we translate from an action scenario description language into a first-order theory with a circumscription axiom. I consider 2nd-order theories to be part of classical logic. From an analysis of the circumscriptive theory, we can identify different means of deriving efficient computational mechanisms for reasoning or "querying" a class of action scenarios.

The advantages we have derived from this approach are the following:

1. A separation of the ontological and epistemological analysis from the generation of computational procedures for querying action scenarios.
2. A direct means of comparing the ontological and epistemological assumptions we make with those of others, including across paradigms.
3. A somewhat less-direct means of employing part or all of particular computational mechanisms proposed by other groups, such as the use of "extended logic programs", regression techniques, or "explanation closure" approaches, for example.

This brings me to the more specific topics of modularity in translation and monotonicity/nonmonotonicity.

As regards modularity:

In our current family of logics (TAL), we find that the circumscription policies used are really quite simple, basically nothing more than predicate completion. This given, we can either reduce the 2nd-order theory associated with an action scenario using a reduction algorithm in a "nonmodular" manner, or generate a "modular" translation directly from the action scenario description which includes one additional formula for each predicate completion. So, I'd disagree with Vladimir's view on the coupling between modular/nonmodular – monotonic/nonmonotonic in his reply. Although one can interpret the use of a reduction algorithm as nonmodular, one can also interpret the use of local syntactic translation as modular. Again, it all comes down to what is meant by "modular" and we may have slight disagreements on pinning down a definition.

As regards the monotonicity/nonmonotonicity issue:

One fascinating insight which the translation into a circumscribed theory provides us with, is that the automated reduction of the circumscription axiom to a logically equivalent 1st-order formula, essentially generates what could be interpreted as "explanation closure axioms". This has been noted and used by other researchers in other contexts such as Vladimir, Ray Reiter, and Fangzen Lin, although in these cases, one works directly with syntactic translations on an existing 1st-order theory rather than direct translation from a circumscription formula.

So this could turn out to be a non-issue in the sense that meta-assumptions of a nonmonotonic character are sometimes left outside a formalism, but guide the way we write axioms or use syntactic translations, or the assumptions are part of the actual formal theory as in the case of using circumscription axioms or default inference rules. There should be a straightforward means of providing formal translation between the two "stances". The discusion would than revolve around what criteria, such as elaboration tolerence or efficient theorem-proving methods, contribute to a particular choice of "stance".

One other related point worth discussion is that if one takes a good look at the diverse approaches being proposed and actually applied in some sense of the word "applied", analysis at the classical logic level shows that

1. the minimization policies are very similar to each other, even across ontological choices. Currently, we are not doing much more than a somewhat "enhanced" form of predicate completion.
2. The language fragment used is generally not more than an "enhanced" Horn fragment.

Not surprising, because we want to develop efficient query mechanisms, be they logic programs or non-standard procedural mechanisms.

This is why I like approaches which relate in a direct manner to classical logic. We can say useful things like:

- "the 'occlude', 'release', and 'noninert' predicates relax an overly strong minimal change policy by importing the 'varied' part of a circumscription policy into the object language. This results in computational benefits due to the resulting simplified circumscription policy actually used. The approach also provides a straighforward technique for modeling non-determinism."
- "filtered preferential entailment and nested circumscription are useful technical tools for increasing the granularity at which minimization can be applied to action theories, but with the added danger of introducing non-consistency preserving formalisms when distinguishing between observations and action occurrences."

These techniques have informal correlates in the original work by McCarthy and have been refined into formal techniques used by a number of researchers in this area. The problem is that this type of analysis is rare. A contributing factor to the lack of generic analyses could very well be the diversity of specialized action languages and the often complex and direct translations to procedural or computationally oriented frameworks.

From: Pat Hayes on 25.11.1997

Vladimir Lifschitz writes:

> *It seems to me that the semantics of an action language cannot be described by specifying truth conditions for its propositions. The problem is the same as with nonmonotonic languages in general. Take, for instance, the closed-world database P(1),P(2). The negation of P(3) is a consequence of this database, but this fact cannot be justified on the basis of truth conditions for P(1) and P(2).*

But it can be justified on the basis of the truth conditions for P(3), which is just as much a proposition of the language as the first two. Nonmonotonic logics are not classically truthfunctional, but they do have truth conditions. The simple language sketched here has the truth condition: P(n) is true iff P(n) occurs in the database.

Pat Hayes

From: Vladimir Lifschitz on 27.11.1997

Erik,

In ENRAC 21.11 you discuss advantages of the F&F approach, in comparison with action languages, and you write:

> *No need to define new languages all the time, and of comparing newly published languages with previously published ones. We stay with the same language, which is sufficiently expressive right from the start to last for a while ...*

I'd like to understand this better. The need to define new action languages arises when we want to describe aspects of reasoning about action that have not been understood in the past. Here are some examples.

1. Ramification constraints vs. qualification constraints. A fact about fluents sometimes allows us to conclude that an action has an indirect effect, and sometimes that it has an implicit precondition. (It functions sometimes as a "ramification constraint" and sometimes as a "qualification constraint.") Example: the objects that I have in my pocket are in the same place where I am. After I come home with a comb in my pocket, the comb will be in my home also; this is an indirect effect. Since knives are not allowed in airplanes, I can't board an airplane with a knife in my pocket; this is an implicit precondition.

2. Asymmetry of ternary constraints. Consider a spring-loaded suitcase with two locks. Its state can be described by three fluents: lock 1 is open; lock2 is open; the suitcase is closed. The constraint is that these fluents cannot hold simultaneously. Consider a state in which one of the locks is open and the suitcase is closed. When I open the other lock, this causes the suitcase to open. (This action does not cause the first lock to close, which, logically speaking, is another possibility.)

3. Interaction between concurrently executed acions. Consider lifting the opposite ends of a table upon which various objects have been placed. If one end of the table has been raised, the objects on the table will fall off. But if both ends are lifted simultaneously, the objects on the table will remain fixed.

These phenomena could not be described in the original action language A and in some of its successors. New, more expressive languages had to be designed. I am wondering what the status of examples 1-3 in the F&F framework is. Would you be able to formalize them in your original language, which you described as sufficiently expressive right from the start? My understanding of the possibilities of F&F is not sufficient to answer this without your help. But about the situation calculus I know that new syntactic features had to be added to it to address these problems, such as the predicate *Caused* (similar to *Holds*, but not quite the same), and an addition operation on actions (to represent concurrent execution).

Patrick,

In ENRAC 21.11 you write:

> *To clarify my "current" working view of the role action languages should play, I simply quote the original intuition Vladimir himself at one time had about their role in his paper, "Two Components of an Action Language":*

> *"Originally, action languages were meant to play an auxiliary role. The primary goal was to represent properties of actions in less specialized formalisms, such as first-order logic and its nonmonotonic extensions, and the idea was to present methods for doing that as translations from action languages."*
>
> *My group currently uses this approach and it is also one of the cornerstones of the Features and Fluents methodology.*

Indeed, your TAL is essentially an action language. There are minor differences in style between TAL and the languages that I've been working on. What you write as

```
[t1,t2] move(p,l) ~>
  [t1] !(place_of(p) == l) -> [t1,t2] place_of(p) := l
```

I would maybe represent this as

```
move(p,l) CAUSES place_of(p)=l,
IMPOSSIBLE move(p,l) IF place_of(p)=l.
```

This is slightly more concise because t1, t2 are suppressed. I am wondering whether you would lose any important expressivity if you changed your macros in a similar way.

Tom,

In ENRAC 21.11 you write:

> *The most difficult issue that I am aware of, is that it is unclear whether*
>
> ```
> A causes F if G
> ```
>
> *means that*
>
> ```
> in every actual state S where G is true,
>       then F is true in R(A,s),
> ```
>
> *or the similar, but different*
>
> ```
> in every possible state S where G is true,
>       then F is true in R(A,s).
> ```

I would say it's the latter.

Pat,

In ENRAC 25.11 you write:

> *Vladimir Lifschitz writes:*
>
> > It seems to me that the semantics of an action language cannot be described by specifying truth conditions for its propositions. The problem is the same as with nonmonotonic languages in general. Take, for instance, the closed-world database P(1),P(2). The negation of P(3) is a consequence of this database, but this fact cannot be justified on the basis of truth conditions for P(1) and P(2).

But it can be justified on the basis of the truth conditions for P(3), which is just as much a proposition of the language as the first two.

Good point. Our closed-world database determines the model in which P(3) is to be evaluated according to the truth conditions of classical logic.

Similarly, a domain description in the language $\mathcal{A}$ determines the transition diagram in which a value proposition is to be evaluated when we want to determine whether it is a consequence of the description.

Further you write:

Nonmonotonic logics are not classically truthfunctional, but they do have truth conditions.

Here I cannot fully agree with you. A default theory in the sense of Reiter is defined by its axioms and its defaults; we have truth conditions for axioms, but not for defaults.

Vladimir

Vol. 1: *156-158*, 30.11.1997
http://www.ida.liu.se/ext/etai/actions/nj/9711-3/

References to Application Areas

Edited by: Erik Sandewall
Linköping University, Sweden

This heading will be used for links to actual or potential application areas where reasoning about actions and change may have a role to play. One of the explicit goals of the present News Journal is to contribute to better links between our area of research and its applications.

[28.11.1997] A while ago we had a plea from Austin Tate and from the present editor about the importance of participating in application areas where methods for reasoning about actions and change are important. The following CfP for a journal issue is interesting from this point of view. Notice that it first mentions research topics in our area, and then it proceeds to invite papers that steer in another direction. Therefore, **although** this particular issue does not focus on actions and change, **still** our area is put forward as highly relevant. Food for thought! - The editor.

CALL FOR SUBMISSIONS

Special Issue of the Journal "Artificial Intelligence in Medicine"

Theme: NET-BASED DECISION SUPPORT IN MEDICINE

Guest-Editor: Lawrence Widman (University of Texas HSC - San Antonio)

Background

The rapid growth of networking, both as the universally accessible Internet and as institutional intranets, has changed one of the fundamental premises of research in artificial intelligence in medicine. Until now, data were often not available in electronic format and professional users would rarely use computers in the course of their daily health care activities. Over the past several years, however, much has changed. Hospitals and large physician practices have installed sophisticated information systems that are truely replacing the paper record and therefore resolve the economic and social barriers to data and to professional users, respectively. Also, the universal acceptance of Web software technology and the promise of Java as a hardware-independent delivery platform have gone far toward resolving technical barriers to the delivery of expert computer technology in the clinical setting. It is fit, therefore, to reexamine the topic of decision support for clinical practice in the context of the new network paradigm that promises to empower both users and software developers.

Many fields of medicine are well-suited as domains for development of decision support systems as their clinical aspects become quantitative, causal, and reasonably well computerized. For example, cardiovascular medicine presents a number of challenges to computer scientists: multilevel causal modeling, reasoning at multiple time scales, reasoning simultaneously in space (anatomic localization) and time, and complexity of interpretation of real clinical records. These challenges obtain in many other areas of medical practice as well. As a practical matter, improved decision support and monitoring tools that improve patient outcomes can reduce health care costs while improving the quality of care. For these reasons, this progress in this area is timely and interesting not only to computer scientists and computer-literate health care providers, but also to health care organizations such as hospital and third-party payors.

Objective of special issue

The objective behind this special issue on "Net-based Decision Support in Medicine" is to report on both applied and reasonable theoretical developments. Applied reports describe implementation and validation of architectures. Because widespread availability of Net connections is so new, reasonable theoretical reports describe prototype systems that explore novel and useful concepts and are therefore worthy of consideration before they are fully validated.

More specifically, papers are expected to cover pertinent topics in networked decision support in medicine. Some examples are:

- Client-server architectures for existing validated expert systems in which one or more design or optimization tradeoffs arose from limitations such as in network bandwidth, computing power in the client computer (e.g., wireless handheld units), interfaces with legacy and/or massive databases.
- Networking issues such as security and/or confidentiality that could not be resolved by straightforward application of asymmetric key cryptography or other standard means.
- Graphical user interface issues relating to the emerging standards of Web browsers, HTML and its extensions, and Java and its variants.
- System design issues arising from interconnection of groups of servers such as, for example, a database server, a UMLS vocabulary server, a computation server, and the client software.

Additional topics will be considered: please send them promptly to the editor (see below) for informal review.

Schedule

All the manuscripts submitted will be subject to a rigorous review process. The special issue will include 4-5 papers of 20-25 manuscript page each, plus an editorial. Manuscripts should be prepared in accordance with the journal "submission guidelines" which are available on request.

January 1, 1998
Submission of tentative title and abstract
to declare intention to submit paper.
Email may be send to widman@sones.uthscsa.edu

April 1, 1998
Receipt of full papers. Three copies of a manuscript
should be sent to:

Lawrence E. Widman, MD, PhD
Division of Cardiology, Room 5.660U
University of Texas Health Science Center
7703 Floyd Curl Drive
San Antonio, TX 78284-7872
widman@sones.uthscsa.edu

May 15, 1998
Notification of acceptance

July 1, 1998
Receipt of final-version of manuscripts

April, 1999
Publication of AIM special issue

ETAI Policy committee

ETAI Area Editorial Committee for Reasoning about Actions and Change

September, 1997