AAAI-08
Main Technical Track

print help

User: Ryan Kelly (rfkelly)
- Start page
- Edit user data
- Logout

Author
- View own papers

Review Feedback and Response

- The purpose of author feedback is to point out technical errors or significant misunderstandings in the reviews; it is not to dispute the opinions of the reviewers, or to explain how criticisms could be addressed in a future revision of the paper.
- Do not necessarily expect replies to your feedback in the final reviews. The senior program committee will ensure that feedback identifying legitimate flaws or misunderstandings will be taken into account in final reviews and decisions.
- Please do not provide feedback explaining how you can address the referees' concerns or criticisms in future versions of the paper. The paper must be publishable as submitted.
- Feedback is strictly limited in length to 150 words per review, no exceptions.

Paper ID	9
Paper authors	Ryan Kelly, Adrian Pearce
Paper title	Cooperative Execution in the Situation Calculus using Prime Event Structures
Paper subtitle	
Paper Type	AAAI-08 Technical Conference Paper (6 page maximum)
Keywords	Reasoning About Plans, Processes, and Actions::Reasoning about Action and Change ** Agents::Multiagent Planning ** Agents::Coordination and Collaboration
Abstract	We develop a powerful extension to the situation calculus for representing and reasoning about the cooperative execution of tasks by a team of agents in asynchronous, partially observable domains. Existing applications of the situation calculus utilize totally ordered sequences of actions, requiring constant synchronization between agents that is undesirable or even impossible in some settings. Our new approach utilizes joint executions, partially ordered branching action sequences based on a prime event structure. We show how a team of agents can use these structures to cooperatively plan and perform an execution of a shared ConGolog program. The use of prime event structures has the capacity to guarantee legal executions in spite of asynchronicity and partial observability, facilitating the use of powerful non-deterministic programming techniques for multi-agent problem solving tasks.
Download	

	Relevance	Significance	Soundness	Novelty	Quality	Clarity	OVERALL	CONFIDENCE
Review	8	6	6	7	5	5	6	4

Comments to author(s)

RELEVANCE: Uses "prime event structures" to coordinate multi-agent execution of a con-golog program. The general topic is of wide interest in AI.

SIGNIFICANCE: Can't tell. See below.

TECHNICAL SOUNDNESS: I don't find obvious problems, but it is difficult to follow the definitions. See "clarity" below.

NOVELTY: Not sure how novel it is to adapt prime event structures to this problem, and to place the approach in the con-golog setting. Seems novel to me.

QUALITY OF EVALUATION: The paper mentions an implementation, and gives a vague idea of how the approach works on an example. It would probably help alot to work out the example in much more detail, and use it to illustrate the *many* layers of definitions, as you go. The paper promises to "show how a team of agents can cooperate to plan and perform an execution of a shared ConGolog program". But it appears that what happens is maybe less interesting, although still useful: there is a central planning system that transforms a multi-agent congolog specification into a "joint execution", which is a kind of partially-ordered plan.

CLARITY: Sentences and paragraphs are well-written. The reader gets a fair sense of the overall approach, but I think it would be rather difficult for a reader to work out all details from what is given.

This may be simply a problem of space. The version of sitcalc employed here is not described fully, and we see no example of successor state axiom using the less familiar constructs of observations and outcomes. The connection between prime event structures and the congolog program is apparently meta-level -- via "histories". No example of a history is given, as far as I see. The presentation and discussion of prime event structures and joint executions is difficult to follow. I take it that figure 2 is intended to serve as a simple example that illustrates joint execution. So I would like to understand how this structure is derived from an action theory (and congolog program?). As far as I see, there are 2 sentences in the text, and a caption, that explain figure 2. The first sentence says "Figure 2 shows a prime event structure specified in this manner (action and outcome events are specific to joint executions, and are defined in a later section)." The second sentence (in a later section) says "Here act1 has two possible outcomes, while act 3 must occur after both act1 and act2." (I suspect that the first "act1" should be "act2"?) I think the reader needs more help than this. There is a sequence of five restrictions that are to be "imposed" on joint executions. It appears that this where is we learn on what basis joint executions are constructed on the basis of the action theory. Perhaps an example involving a simple joint execution something like figure 2 could be worked through in some detail, so that a reader has a better chance of following the definitions?

REBUTTAL QUESTIONS:

You may optionally submit feedback that will be visible to the reviewers and PC members. The length of your feedback is limited to 150 words (including spaces and special characters).

Please enter your arguments here:

Back

Submit

Review	Relevance	Significance	Soundness	Novelty	Quality	Clarity	OVERALL	CONFIDENCE
	8	4	6	4	2	4	4	7

Comments to author(s)

RELEVANCE:
This work is in the area of temporal reasoning and planning. It focuses on a central area of knowledge representation. It is particularly relevant to those who use the situation calculus and ConGolog, but should be relevant to anybody working int he fields of temporal reasoning and planning. The work is relevant to core concerns in the AI community.

SIGNIFICANCE: The paper is only moderately significant. The title and abstract give the impression that it is more significant than would be warranted by the results in the paper. The paper really isn't about "cooperative executions" and "joint executions" in the sense that we'd normally use these terms. The agents aren't reasoning about the consequences of their plans, or how their actions will interleave, or how they are interacting with other agents. Rather, the paper is really just about concurrent actions and sharing of resources, a very different topic. Others have taken the notion of cooperative and joint executions further (see below under novelty) which detracts from this paper's significance.

In general, the results presented are quite weak. The theorems presented (in the Independent Actions section) are rather trivial. There is a nice development of the notion of restrictions on joint executions (pp. 4-5), but these are discussed in a vacuum. How are these restrictions enforced? Are there any formal results relating these restrictions to properties of the ConGolog programs that obey these restrictions?

TECHNICAL SOUNDNESS:
I did not find errors in this paper. However, as mentioned above, there are few results, so it's difficult to evaluate. Also, the full example is never written out, neither in the situation calculus nor in ConGolog.

NOVELTY:
There has been other and deeper work in this general area. The authors themselves mention related work in the specific area of ConGolog, such as the work of Shapiro, Lesperance, and Levesque, 2002, and of Farinelli, Finzi, and Lukasiewicz, 2007. The authors argue that their work overcomes limitations of such previous work. For example, they say that the system of Shapiro et al. cannot handle partial observability of actions. The difficulty with this claim is that the authors do

not show that their system can handle partial observability of actions. All they do is say that they should be able to handle it because they don't require the full situation term. However, to support the claim, the authors should give an example in which the jointly executed plan includes partial observability of actions. Similarly with respect to their claims regarding Farinelli et al: the authors claim that the Farinelli approach requires explicit synchronization through communication and a shared state while their approach uses the actions observed by each agent. Again, this claim is not supported in the example, in which there is no mention of the observation action. (Though, it is hard to tell, since there is no formal write-up of the example.)

Outside of the situation calculus, deeper work in this area, include the work of Davis and Morgenstern, 2005, which considers how an agent can delegate a plan to multiple agents working on various pieces of the plan. A protocol for correctness is given which ensures, among other things, that resources are allocated appropriately. Their system has restrictions that are significantly less burdensome than the ones given in this paper's Restrictions on Joint Executions section. The theory is much more powerful in that agents reason about other agents' knowledge, abilities, etc.

QUALITY OF EVALUATION:

This is a formal KR paper, so I don't expect the standard AI paper "evaluation" section with lots of bar graphs and statistics. Still, I would would have expected the following:
1. A description, as precise and formal as possible (though not artificially so), of the classes of problems that could not be handled with previous approaches, but that could be handled with this approach.
2. A description, as precise and formal as possible, of the classes of problems that still cannot be handled with this approach.
3. A detailed formal write-up of the actual example. What is the exact input? (We only see the output, and only an informal approximation of the output.) There probably isn't space for this, so a pointer to supplementary material would have been nice.

The absence of 1. and 2. led me to give the low rating on evaluation.

CLARITY:
The general writing style is good. However, as a whole, this paper was hard to read. This was due mostly to not explaining the terms and concepts that were used, and the choices that were made. For example, "outcomes" are defined as sets of events, though one normally thinks of an outcome as a state of affairs. What was the reason for this choice? What is the intuitive notion of "covers" in the usage outcome set e1 covers outcome set e2? What are some examples?

REBUTTAL QUESTIONS:
Define the class of problems which your theory can handle which other theories cannot handle. Give examples.
Define the class of problems which your theory cannot handle. Give examples. What would need to be added?
Do you have any formal results on the connection between the restrictions on joint executions, the ConGolog programs you write, and the output you get?

You may optionally submit feedback that will be visible to the reviewers and PC members. The length of your feedback is limited to 150 words (including spaces and special characters).

Please enter your arguments here:

Back Submit

Relevance	Significance	Soundness	Novelty	Quality	Clarity	OVERALL	CONFIDENCE
7	4	6	6	4	4	4	8

Review

Comments to author(s)
RELEVANCE:

ConGolog and the situation calculus are relevant; their execution is a relevant topic as well.

SIGNIFICANCE:

The motivation of the paper is clear (in the broadest terms) but the significance of its technical contributions is not. The idea seems to be that we would make the situation calculus support concurrent actions because of the environments where it might then be applied (or is being applied).

Thus prime event structures (from concurrency theory) are used to represent joint executions.

But the situation calculus wasn't quite designed for concurrency and the solution comes out rather artificial. There are at least two major shortcomings, which render the approach quite useless for the kinds of needs this reviewer can imagine.

1. Although joint execution is allowed, the idea still is that there will be a universal, apparently centrally created, plan that describes the actions of all agents.

2. The theory ends up restricting joint actions to the point where the potential enhancements are all lost. In particular, the authors impose five restrictions. Of the first one they state that it is "key to the power of joint executions". This restriction is that if actions may be performed in either order then the actions must be independent (in the sense of being possible and having the same outcomes in either order). This would be reasonable as a well-formedness condition on plans that may be executed in a distributed fashion.

However, in the present setting, it is stated as a constraint on the representations. It means that the agents have no autonomy. The agents can only execute centrally created plans and nothing else, and the plans would ensure that they act concurrently only when the concurrent actions are independent.

TECHNICAL SOUNDNESS:

The paper appears sound for the most part. There seem to be some errors, which however are of the kind that could be corrected.

The definition of ens(i) is not clear.

Say we have x \preceq y and y \preceq i. Then ens(i) cannot be {x, y} because x \preceq y. However, ens(i) would have two solutions. It could be {x} or it could be {y}, each of which is maximal as the definition requires. I believe the authors wish it to be {y} and not {x}

The above has a consequence on the definition of alts as well, for which erroneous or unintuitive examples can also be constructed.

Why does R3 refer to ens(i), which are events that precede action i, although it talks about the outcomes of the action i?

NOVELTY:

The paper applies event structures to model histories as in situation calculus.

QUALITY OF EVALUATION:

There isn't an evaluation as such but the approach is applied on an example.

CLARITY:

I found the paper difficult to read and understand. There wasn't adequate motivation at a technical or a conceptual level.

REBUTTAL QUESTIONS:

You may optionally submit feedback that will be visible to the reviewers and PC members. The length of your feedback is limited to 150 words (including spaces and special characters).

Please enter your arguments here:

Back Submit