Comment for Paper: Validating the Autonomous EO-1 Science Agent

Werner Stephan
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3, 66123 Saarbrcken, Germany
stephan@dfki.de

1 Summary

The main topic I would like to put forward for a discussion is the distinction between model building and the generation of (plan) sequences that are consistent with (or correct with respect to) the basic model. This distinction seems to have severe impacts for the main topics discussed in the paper:

- the design process for autonomous systems,
- hazard analysis and safety mechanisms, and
- quality assurance, i.e. validation and verification

In particular these aspects have to be discussed in the context of the layered architecture of the underlying system that gives rise to different levels of abstraction.

2 Design Process / Model Building

As described in the paper the design process requires intensive cooperation with (domain) experts (analysis of previous manually constructed solutions, reviews etc.). To my understanding this expert knowledge goes into the basic model that underlies the planning process. In particular this concerns parameters and constraints. With respect to model building it seems worthwhile to consider the following remarks / questions:

- A few more comments on how the model is structured and described would be useful to answer/discuss the question: What is given (assumed) by the model and what can be inferred as properties of the model?
- Would a special presentation of such models (view) be useful to communicate with domain experts?
- Does the model cover unexpected events caused by failures or influences from outside?
- Does it seem reasonable/feasible to have a formal (possibly abstract) model of the relevant aspects of the space craft and its instruments. Such a (deep) model would allow to define a semantic interpretation of plans and command sequences.
- In what sense could the current model be called “formal”?
- If not, what are the main obstacles for a deep formal model? Is it the need to incorporate physical aspects (discrete versus continuous changes)? Has it to do with the “unexpected events”? Is it the complexity of the system?
- Otherwise, what would be the adequate (formal) description technique?
- With respect to model building the step from the CASPER level to the SCL level looks like a kind of refinement problem. It would be useful to comment on precise relation between these levels of abstraction. For example, where do the additional details in the example scripts come from? As for a (more) formal treatment, would it be possible to come up with a notion of correct (or safe) expansion?

3 Hazard analysis / Safety Mechanisms

¿From my personal view the main question is: What hazards are (can potentially be) analyzed within a model? Was there a (formal or informal) notion of a safe behavior that can be checked (tested) or even proved?

This would allow to come up with some notion of correctness for safety mechanisms in the sense that unsafe states (that have to be part of the model) are ruled out.

The layered architecture relies on multiple protection mechanisms. Was there some kind of analysis what safety properties can be guaranteed by construction (of safe plans) at the upper level? If, in the future, planned sequences can be (formally) proved correct with respect to certain safety properties (like the maximum time of operation), what would be the impact on lower level safety filters.

4 Quality Assurance

As already indicated at the beginning of section four the most important distinction is between

- the correctness of planned sequences with respect to a given model (including correct refinements if several layers of abstraction are considered) and
- the adequateness of the model itself.

It is not entirely clear to me whether in real world applications there really can be a sharp borderline between what

could possibly be called verification (possibly incomplete if done by testing) and validation.

As far as verification is concerned a (more) formal model could potentially justify uniformity assumptions in the generation of test cases.