Commentary on:
"Massaging the Plan with the Language for Mission Planning"
by Bruno Teixeira de Sousa

Mark D. Johnston
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena CA USA 91109
mark.d.johnston @jpl.nasa.gov

Introduction

This paper describes a "Language for Mission Planning", LMP, that has been developed for use by the Venus Express (VEX) Flight Control Team. LMP allows the operator to define "logical, linear temporal rules that can be systematically applied over the data elements of a plan to verify relations among them and to generate new activities or modify existing ones". Among the benefits provided by LMP are:

- rule-based
- organized in an editable logical workflow
- flexible and easily programmed by an operator
- unambiguous and predictable

LMP is declarative in nature, so the user specifies "what" is desired but not necessarily "how" the system should accomplish it. The paper provides some examples of LMP rules dealing with determining VEX spacecraft communications periods, as well as inserting transponder on/off sequences.

LMP has met with considerable success in the VEX domain, and also used in the EMS (ESTRACK Management System) and Mars Express systems. One of the advantages noted is that LMP rules are managed by the Flight Control Team with "no software development needed", with a resulting rapid turnaround for changes. There are plans to apply LMP to a number of future missions.

Comments and Questions

It would be interesting to have some further discussion of some points mentioned briefly in the paper, for example:

- **Rule-based** systems have been known to have some drawbacks where the unexpected interactions of large rule sets can make maintenance difficult (e.g. it can be very hard to make rules modular). For example, what determines rule execution order? Suppose you have a rule to insert on/off commands, and another to check if they are too close together. How can you be sure that the "too close" check is always guaranteed to run after all the insertions are done? Can there be rule cycles?
- **Rapid turnaround usually competes with operational risk** and it would be interesting to hear how this trade-off is handled in the development and validation of LMP rules. For example, what assurance is there that a rule written by a FCT member and put into production will be safe for the spacecraft? What about as the size and complexity of the rule set grows?
- **Comparisons with other execution languages** and scripting languages would be very interesting. Although it is noted that LMP does not require software development expertise, the features of the language do not seem too far removed from other languages. Languages like Python have been adapted for use in specialized domains like numerical analysis and image processing.
- **Prospects for extensibility** would be interesting to hear discussed. How might LMP incorporate search and optimization? Could custom LMP operators be developed that would integrate with existing AI systems and provide access to their modeling and optimization capabilities?