Commentary on The RAXEM Tool on Mars Express – Uplink Planning Optimisation and Scheduling Using AI Constraint Resolution

Commentator: Reiko Tsuneto Rager
Space Telescope Science Institute
3700 San Martin Dr.
Baltimore, MD 21043, USA
rager@stsci.edu

Introduction

The paper describes REXAM, an uplink planning tool for ESA's Mars Express (MEX) mission. The tool helps the MEX operators plan their uplink command schedules. It automates the process which was previously manual, speeding up the time for planning while obeying all constraints. It is a decision-support tool that gives more control and options to users.

Problem Domain

The paper describes a domain with some features that make the problem interesting:

- The uplink window duration varies, which constrains the amount of the telecommands it can uplink.
- The amount of telecommands to be uplinked is further limited by how much memory is available on-board at the time of the uplink.
- A command sequence always has to leave the spacecraft and instruments in safe states. This means the tool cannot cut the command packages anywhere it wants to fit them in the available uplink windows.

There are also rules on which commands to uplink first, if full confirmation is necessary or not, and if the backup uplink windows are needed or not.

Approach

The tool was developed by the same team who developed MEXAR, the downlink planning tool for the Mars Express. There are similarities in the problem domains between MEXAR and RAXEM and both were coded in Java. However, the paper is not clear about if and what part of MEXAR code was re-used in RAXEM. I also wonder if the 'look and feel' of the MEXAR GUI was kept in the RAXEM GUI?

The authors claim RAXEM employs 'constraint resolution' technique. Unfortunately, few details can be found in the paper about the kind of algorithms and techniques used. The paper states the focus of the paper is not on the technical details. Nevertheless, I feel a brief technical description would be useful. For example, does it use the same or similar AI techniques used in MEXAR?

Traditional AI systems tended to be black-boxes. The user puts in all necessary inputs and the system outputs a solution without much explanation. Recent AI systems are more white-box, mixed-initiative systems where the system provides multiple choices and interacts with users to guide the search. RAXEM seems to be one of the latter, which I think is very important in winning the trust from the users. In my experience with the Hubble Space Telescope, I have noticed that many times, scientists, engineers and operators do not just accept the software output without question. We had to make our software to be able to provide additional information when requested, so the users can figure out how to get the output they want. RAXEM provides user-controlled decision parameters to guide the solution and a GUI to show how a solution plays out on-board.

The usability of software is an important aspect in deployed systems. Very often, software requirements focus on the functionality and ignore the usability. The authors state RAXEM was built based on concise requirements. Did the requirements specify the user-interface? Is RAXEM integrated well with other tools used in the planning process? And how has the operational experience of the user-interface been?

Conclusion

RAXEM is a yet another successful deployment of a planning and scheduling application. It is clear that the development team made an effort to reduce the users' resistance to the new tool, by keeping the existing file formats, slowly introducing the system to the operations, and getting user feedback which is immediately addressed. Communicating with users early and throughout the development process helps remove the gap between the users' needs and what the developers envision. We can learn from their success on these aspects.