

Sequence System Building Blocks: Using a Component Architecture for Sequencing Software

Barbara A. Streiffert^{*} and Taifun O'Reilly[†]

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

Over the last few years software engineering has made significant strides in making more flexible architectures and designs possible. However, at the same time, spacecraft have become more complex and flight software has become more sophisticated. Typically spacecraft are often one-of-a-kind entities that have different hardware designs, different capabilities, different instruments, etc. Ground software has become more complex and operations teams have had to learn a myriad of tools that all have different user interfaces and represent data in different ways. At Jet Propulsion Laboratory (JPL) these themes have collided to require a new approach to producing ground system software. Two different groups have been looking at tackling this particular problem. One group is working for the JPL Mars Technology Program in the Mars Science Laboratory (MSL) Focused Technology area. The other group is the JPL Multi-Mission Planning and Sequencing Group. The major concept driving these two approaches on a similar path is to provide software that can be a more cohesive flexible system that provides a set of planning and sequencing system of services. This paper describes the efforts that have been made to date to create a unified approach from these disparate groups.

I. Introduction

The tasks of ground operation teams contain many competing elements. The goal of space science is to increase the knowledge that is known about our solar system. In order to achieve the goal space scientists want to maximize the number and quality of the observations that they make in order to garner the greatest amount of significant data from which to draw that knowledge. The goals of spacecraft engineers are to build more and more capable spacecrafts and at the same time to keep them safe. Ground operations teams must merge those goals and send appropriate sets of commands. However, ground software has become more complex and operations teams have had to learn a myriad of tools that all have different user interfaces and often represent the same data in different ways. Fortunately, the advent of new approaches to building software allows solutions to their hectic environment.

The goal of software has always been to make using computers and their various features easy for users/operators. In today's software world it is possible through the use of various utilities, components, frameworks and rich client platforms to unify the ground software. Two different groups at Jet Propulsion Laboratory (JPL) have been working on tackling the problem of having many tools with different approaches, different representations and differing data requirements. One group is working for the JPL Mars Technology Program in the Mars Science Laboratory (MSL) Focused Technology area. MSL is the next rover going to Mars. It will be launched in 2009. The other group is approaching the problem from the JPL Multi-Mission Planning and Sequencing group. This group creates multi-mission software for use throughout JPL.

II. Next Generation Uplink Planning System (NGUPS) Team

The Next Generation Uplink Planning System (NGUPS) team is part of the MSL Focused Technology group. MSL has decided to inherit the ground system from Mars Exploration Rover's (MER) as their baseline. However,

^{*}Software System Engineer, Section 317, M/S 301-250 D 4800 Oak Grove Dr., Pasadena, CA 91109 USA.

[†]Cognizant Design Engineer, Section 317, M/S 301-250 D 4800 Oak Grove Dr., Pasadena, CA 91109 USA.

several MSL driving requirements meant that the MER ground system tools would need to change in order to meet those requirements. The goal for this team is to explore new technologies in an effort to streamline that baseline. The NGUPS team works closely with another focused technology group, the Ensemble team, and is responsible for integrating the work of the Ensemble team with MER legacy tools. Initially the NGUPS team performed a study of MER operations and the tools used by them. A report was published on the findings of the study. Several results from the study showed areas that needed to be addressed. Members of operations teams wanted a more cohesive set of software tools. Another key element that was missing in the MER ground system was the ability to reuse existing spacecraft activity sets that had been expanded to spacecraft commands.

Using the study as a set of requirements, the NGUPS team looked at each tool in the set to see if there was overlap or redundancy as well as defining the interfaces among the tools. Then by using an open source runtime and development environment, called Eclipse, and refactoring other legacy tools, they hoped to create a system that presented a single environment to the user although in the background different tools and collections of components operated together to provide the necessary planning and sequencing data needed by operations teams. Initially, they started with the MER ground system tools.

Each software tool in the MER ground system has specific tasks. The following is the list of primary tools used by MER and their primary function. The names of the actual tools have been changed to more generic functional names.

- 1) Activity Resources and Targets: A visualization tool used for science planning and resource summary information.
- 2) Constraint Propagation: Temporal and State Constraint specification and checking.
- 3) Activity Planning: Scheduling of activities and activity resource modeling
- 4) Activity Expansion/Command Editing: Processing of activities and expanding them into their commands and editing of command.
- 5) Mobility/Arm Sequencing: Driving the rovers and checking the rover arm placement and movement.
- 6) Sequence Modeling: Modeling the effects of the commands and flight rule checking.
- 7) Interactive Timeline and Table Editing: In MER this function was coupled with each of the above tools. There was no collaboration among the tool creators so that each user interface presented information differently including information that is common to all the tools such as "time".

The NGUPS team started with this set of tools and is combining them into a workable ground system. In the planning area the Ensemble team (as was stated above) based their development on an open source Rich Client Platform (RCP) called Eclipse. The science planning tools including the Activity Resources and Targets tool and the new combined User Interface are fully integrated into Eclipse, and they have created both sharable components and application specific components. In addition, they use the Eclipse workbench as well as creating the user interface in the Eclipse Standard Widget Toolkit (SWT). The other tools are the legacy tools and these tools have been refactored in such a way that they utilize the Eclipse bridge capability to connect to the science planning tools without the need for scripts as was required for MER operations. Except for the Activity Expansion/Command Editing tool all of the tools use the User Interface built by the Ames Research Center Contingent of the Ensemble Group.

Figure 1 shows the new approach and the "plug-in" connections. It is important to note that the plug-ins perform a variety of functions. Some of the plug-ins are very simple and only pass data from one tool to another, other plug-ins are more complicated and do a myriad of tasks like start processes, check status, receive and send data, etc. The last group of plug-ins is tightly coupled with the Eclipse RCP. The one area that is still under development is the way to utilize external models in a consistent, non-redundant way. Those areas are shown in the diagram, but haven't been addressed in a significant way at this time. They are areas that are still in work.

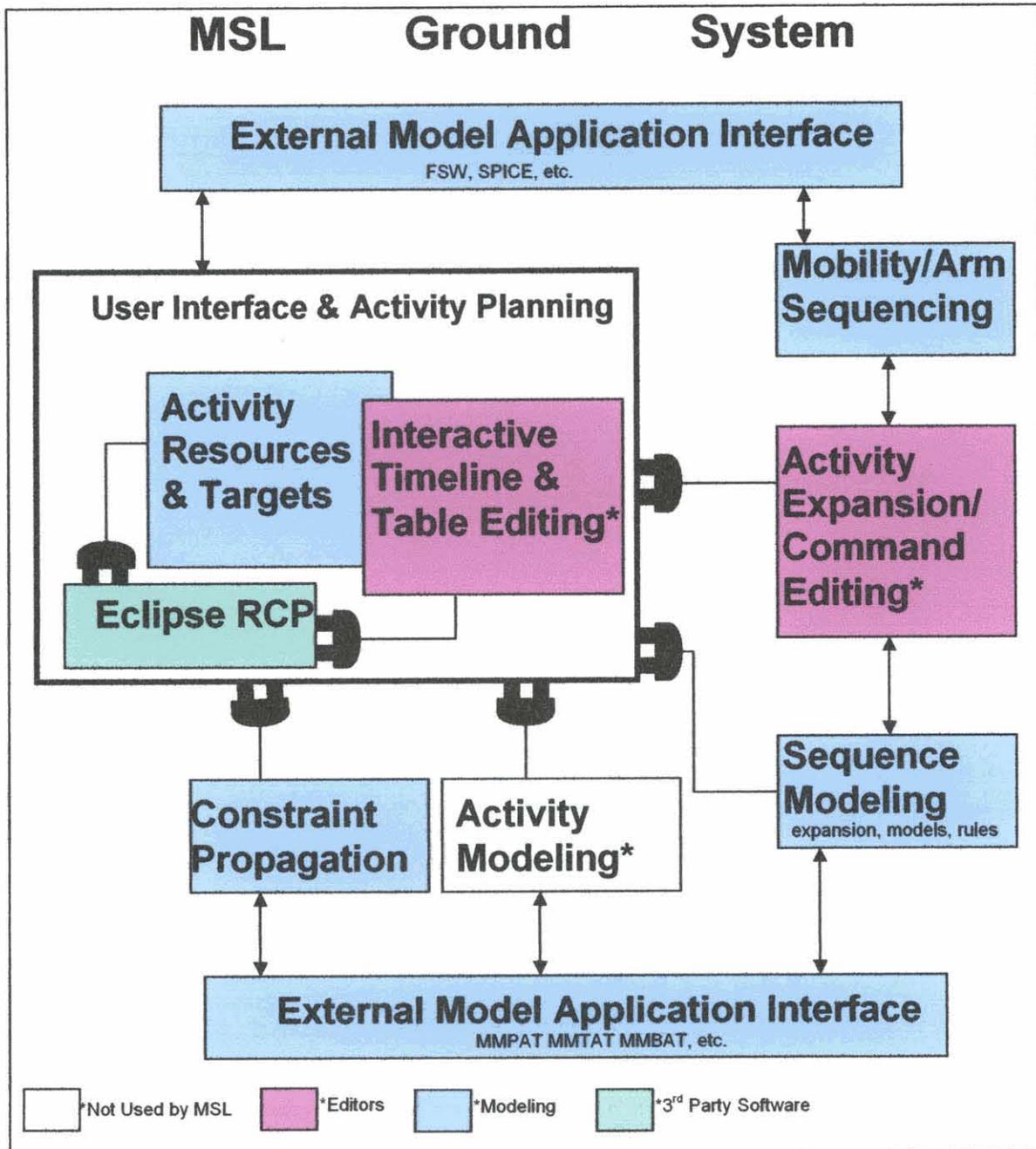


Figure 1 shows the MSL Ground System Software Tools and their connection with each other.

III. Multi Mission Planning and Sequencing Team

The Multi Mission Planning and Sequencing (MPS) team creates multi-mission tools that can be used for any mission. Generally these tools have been created in such a way that there are two pieces – a multi-mission application independent piece called “core” and a mission dependent piece called “adaptation”. Each project takes the “core” tool and “adapts” it using the project specific spacecraft activities, spacecraft commands and flight and mission rules. Adaptation can be in the form of utilizing an interpretive language specific to the tool or in code that is attached via a user defined library. Many missions at JPL and elsewhere use these tools for their ground system.

At the same time that the NGUPS team has been working on this MSL specific ground system, the Multi Mission Planning and Sequencing Team (MPS) has been working to refactor a number of legacy tools into more usable services. In particular they have been working on the Sequencing Modeling tool and the Activity Modeling tool. The Activity Modeling tool has been refactored into an activity modeling server and a user interface client. In this way the software still had the ability to use the older user interface or to have another user interface client attach to it. The Sequence Modeling software took a different approach. This software runs without a user interface most of the time. For Sequence Modeling software it was decided to create a web service so that the software would find and attach to the correct version and perform load balancing without user’s making these determinations. In addition, the software would create its own configuration file from the input of a user or another software element. Finally, a database was attached to this software.

Figure 2 shows the work that is currently being done in the Multi Mission area. The refactoring of the Activity Modeling software has allowed that software to be able to utilize other User Interface software. The Sequencing Modeling changes allow that software to perform more of the behind the scenes aspects alleviating drudgery work for operations teams. The original capabilities are still available in more conveniently packaged into services.

As work progressed, the teams started collaborating. Much of what the Multi Mission team was doing was needed by MSL. Many of the improvements implemented by the MSL focused technology teams were elements that would enhance the capabilities of the Multi Mission team. A proposal to work on a multi mission rover ground system was made and accepted. The In-Situ Collaboration Team was formed.

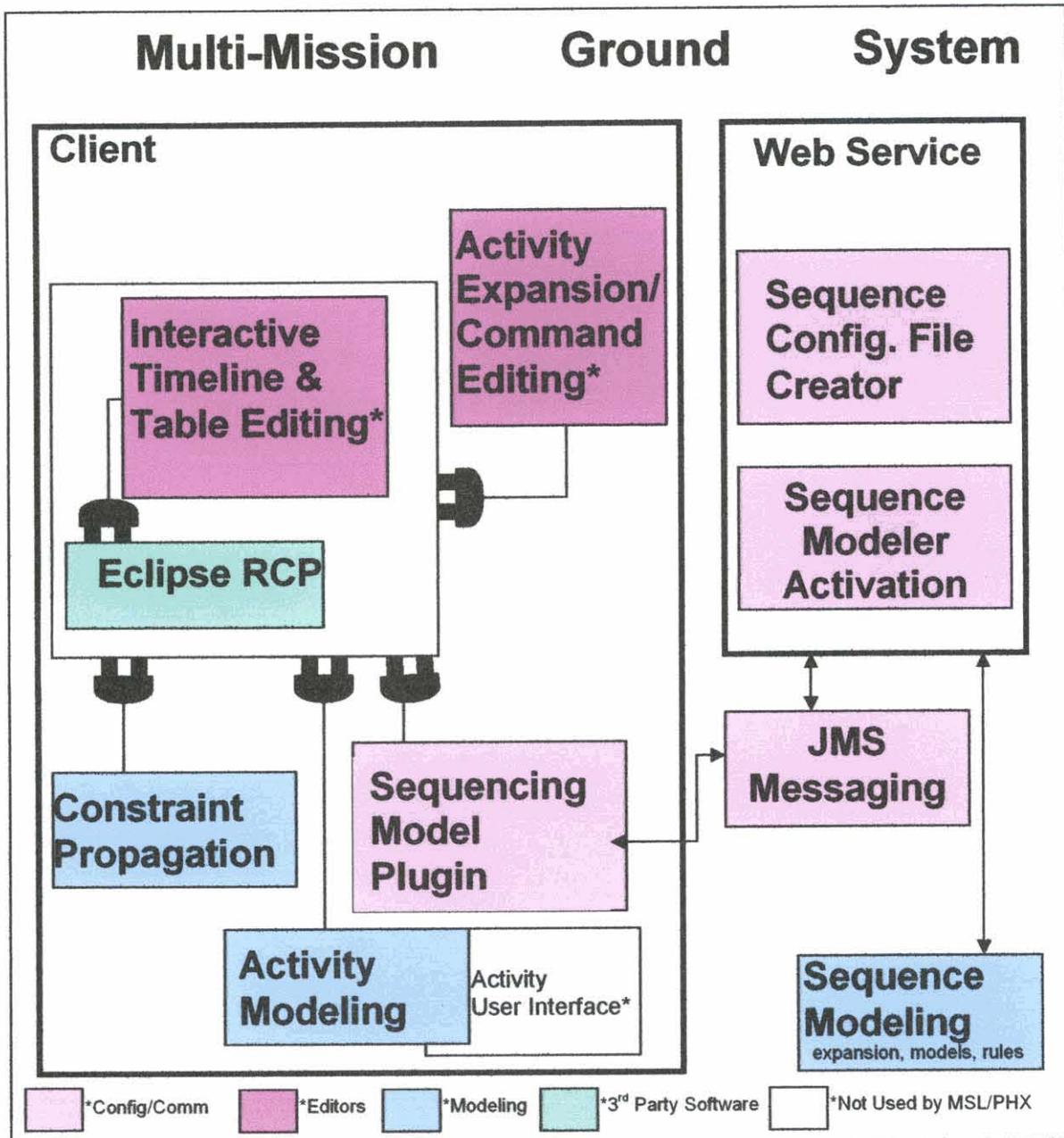


Figure 2 shows the work that the Multi Mission Team worked on to refactor two of its legacy systems.

IV. In-Situ Collaboration Team

The In-Situ Collaboration Team consists of multiple teams. Three of the teams have already been mentioned (the NGUPS team, the Ensemble team including the Ames Participants and the Multi Mission Planning and Sequencing team). In addition, the Phoenix project ground system has joined the team. Phoenix (PHX) is a lander that will operate on Mars and will be launched in 2007. The multiple teams are putting together a ground system that will work for both MSL and PHX as well as for the Multi Mission team.

Figure 3 shows the end result of the collaboration. All the tools will communicate together without scripting and there will only be two editors among the five tools – the activity timeline and table editor and the command editor. The operators will be presented with a single way of representing information. In addition, because of the linking of the tools through plug-ins and other interprocess communication capabilities, sequences of commands will be able to be reused as is or edited as needed. This last capability is a required feature for MSL to meet its schedules and something that is highly desired by PHX. Finally, the ground system is flexible. Any of the software tools can be swapped in or out. MSL is not planning on using the Activity Modeler software and PHX is only planning on using the Constraint Propagation software in a very limited way. The Activity Resources and Targets software and the Interactive Timeline and Table Editing software have components that can be combined in various ways to meet the needs of different projects.

One of the first aspects to be defined in this collaboration has been to identify the work that belongs to each team. Figure 4 gives an indication of the types of elements that are the responsibility of the three collaborators. The list is meant to give a general idea of the type of work each team is doing and not meant to be an all encompassing list. Each team has set out on its set of tasks. As a progress report a demonstration of the state of the software was presented in October 2005 and was extremely well-received. Another demonstration of the software will be presented this summer (2006).

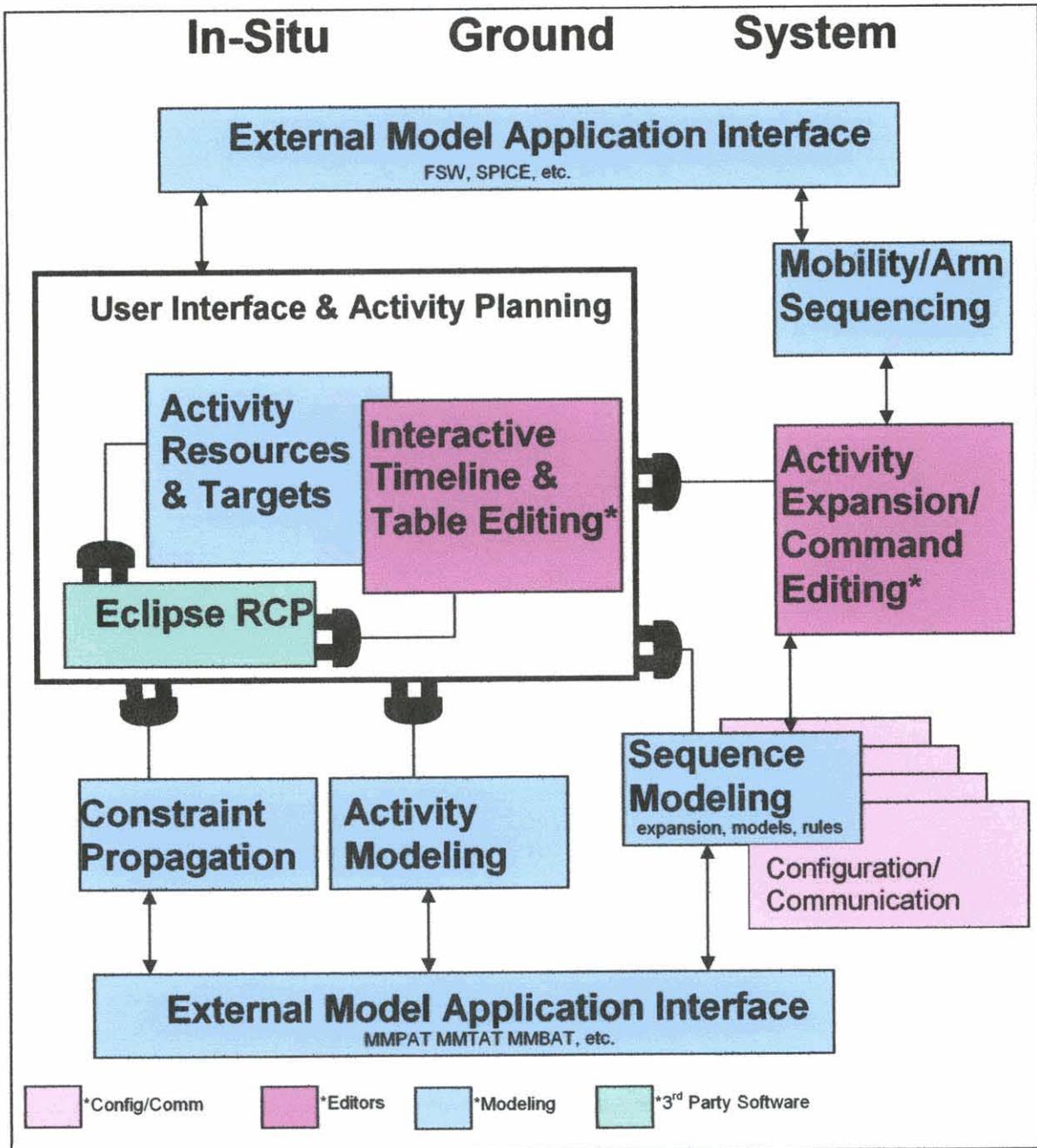


Figure 3 shows the combined ground systems being developed by the In-Situ task for MSL and PHX.

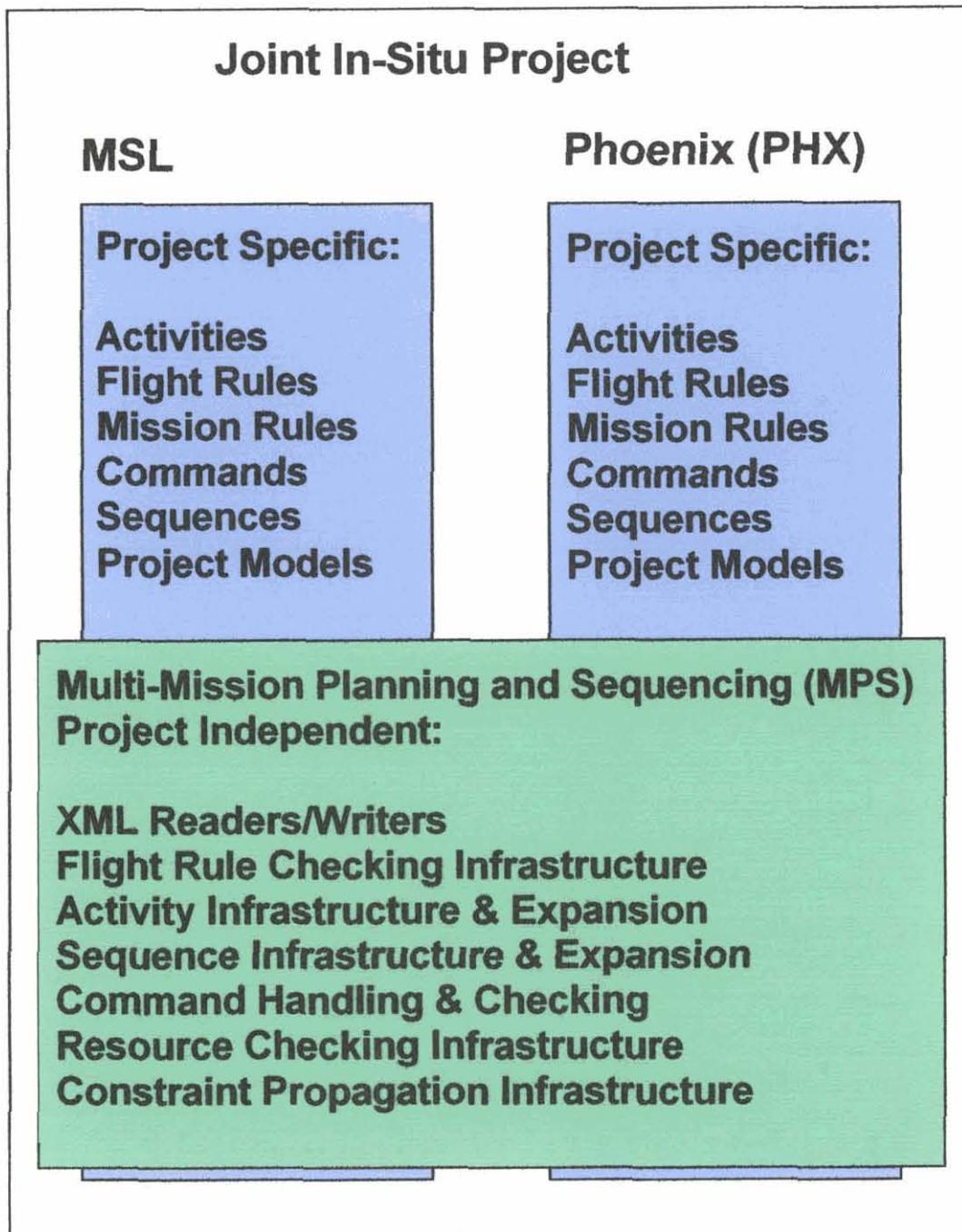


Figure 4 shows the responsibilities for the teams participating in the In-Situ task.

V. Conclusion

Even though the teams are funded by different organizations (MSL, PHX and MPS) and each team has a different lead System Engineer, the approach has been successful. The success of the In-Situ Collaboration Team has come down to the people involved and their willingness to collaborate, communicate and coordinate to obtain the single goal of producing a consistent, usable ground system that enables a productive operations team. Each of the teams knows that the ultimate goal is for future projects to be able to shop for the ground systems tools that they need and put those tools into their shopping cart (Fig.5).

Acknowledgments

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

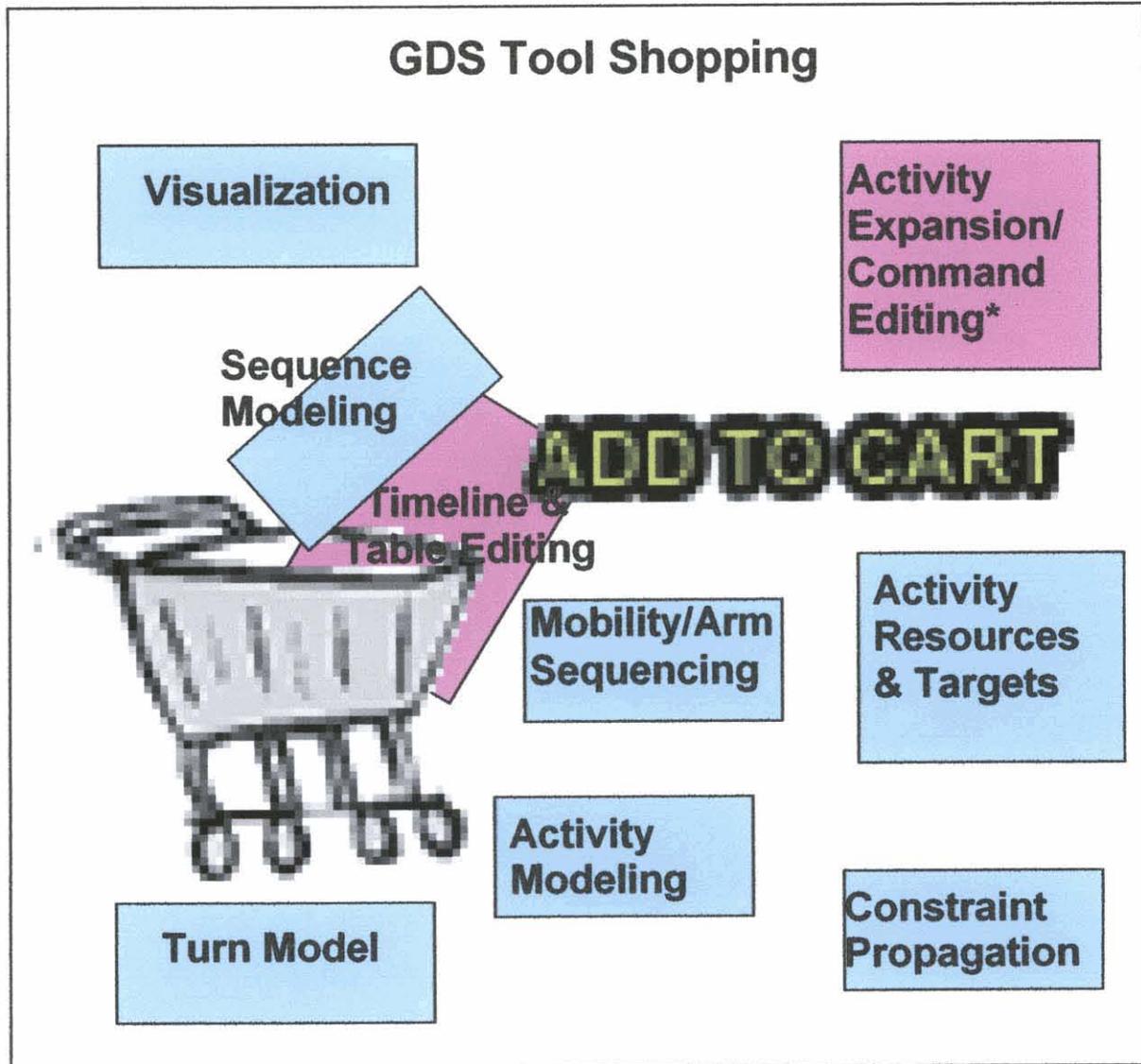


Figure 5 shows the preferred way of selecting ground system tools.