Scheduling and Planning Interface for Exploration (SPIFe)

Arash Aghevli and **Alfredo Bencomo**
Autonomous Systems and Robotics Intelligent Systems
SGT, NASA Ames Research Center
Moffett Field, CA 94035

Michael McCurdy
Human-Computer Interaction Group
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

Many planning tools developed as user-facing interfaces to automated planning systems do not allow users enough flexibility to explore plans in a number of different ways, quickly understand complex sets of constraints and their implications, or experiment with different solutions without fear of losing work. Typically, such tools are architected in such a way that the user interface is integral to the underlying planning, scheduling, and simulation engine(s). The Scheduling and Planning Interface for Exploration (SPIFe) is an integrated planning and scheduling toolkit based on hundreds of hours of expert observation, use, and refinement of state-of-the-art planning and scheduling technology for several applications within NASA. It was designed from the ground up with the needs of the operational user in mind, and it presents unique solutions to a number of problems common in other commercial and homegrown systems. SPIFe has been used on the Mars Exploration Rover mission and the Phoenix Mars Lander mission, and is now being baselined for use on the next Mars Science Laboratory mission (fall of 2011). It has also been adapted as preflight planning and a real-time analysis console tool that supports all phases of planning on the International Space Station (ISS), as well as several other flight projects and analogs.

User Interface Principles and Components

The SPIFe user interface is designed to be a highly adaptable and user-customizable framework for viewing and manipulating plan and schedule data. In order to achieve this, SPIFe employs a composable, plug-in architecture based on the open source Eclipse Rich Client Platform (RCP). Eclipse provides a robust plug-in framework, and the RCP provides many fundamental user interface components, such a tabbed ”workbench” that allows users to manipulate views and editors to display the information most relevant to the task at hand. The following sections describe a number of SPIFe views and editors that can be combined (or omitted) depending on the needs of a particular planning application.

The Timeline

One of the central components of the SPIFe framework, the timeline (Figure 1) provides a traditional time-based representation of a plan. Activities appear as bars that vary in width according to when they're scheduled. Timeline rows are highly configurable: which rows are displayed, their ordering, bar figure look and feel, and row criteria (which determine whether a given activity appears on a given row) can all be modified in a descriptor file for a given application.

The goal of this extensive configurability is to provide an appropriately detailed representation of a complex schedule that is responsive to the needs of a particular user – suppressing details that are not likely to be relevant or understood and presenting others at a level of abstraction appropriate for a user to make planning decisions on his or her own, or easily understand the results of an automated planner. For example, a typical Mars rover plan may include hundreds of individual activities, each with copious metadata – parameters, notes, and results of resource estimates. A user may choose to use a hierarchical grouping to associate activities of similar scientific intent or using similar spacecraft hardware, then work with these higher level activity groups on the timeline when making planning decisions or presenting the schedule to other stakeholders.

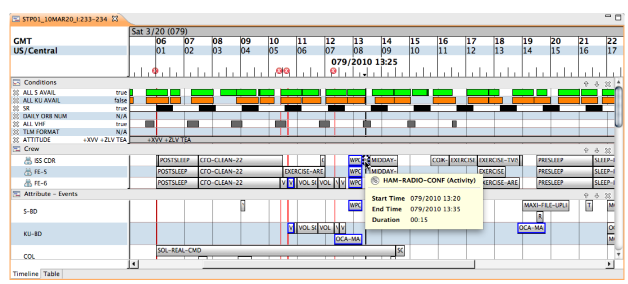

Figure 1: The SPIFe Timeline

All activities can be edited directly via drag-and-drop, and the timeline also provides several feature such as multiple selection, feedback during editing operations, and full support for multiple levels of Undo and Redo to allow users to freely explore multiple solutions. The goal of the SPIFe timeline is to capitalize on user familiarity with common visual editing paradigms where possible (e.g. manipulating figures in drawing tools like Visio or Powerpoint) in order to remain approachable by non-experts.

Internal SPIFe constraint checkers as well as external systems can provide detailed temporal violations and have them displayed within the context of the timeline. Additional information such as violation culprits can be identified visually via activity borders. Tooltips can be enabled to show a configurable level of greater detail on violations or the activities themselves, as well as providing quick access to common fixes for temporal constraint violations or other schedule defects.

The Table Editor

In addition to the timeline editor, SPIFe provides a tabular representation of the activities and groups in the plan. The Table Editor (Figure 2) is useful for displaying a large number of activities, and is especially useful for plans that are sparsely populated (few events over long periods of time) where a timeline display would be mostly empty for a given time range. The Table Editor can be configured with columns representing each piece of activity metadata, including basic start time and duration information as well as details of resource requirements or per-activity resource consumption predicts.

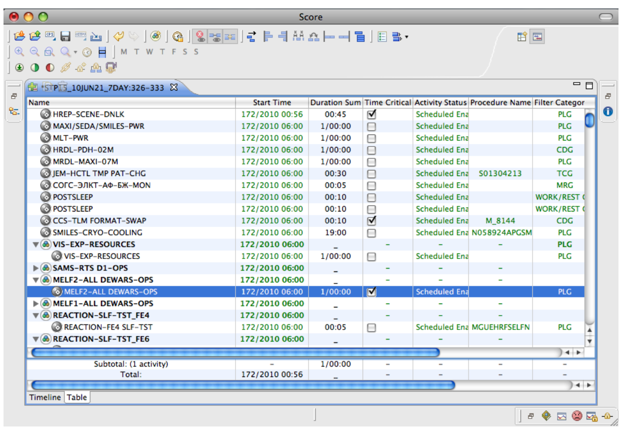

Figure 2: The SPIFe Table Editor

The majority of observed planning processes involve some form of plan integration. For example, the science team on a Mars mission may be broken up into theme groups around instruments or areas of scientific intent. Each team may build partial plans in parallel, and then feed them back in to an integrated plan for the spacecraft. In order to facilitate this merging process, users can open as many plan fragments as needed and simply drag and drop or copy/paste from one editor to another. For more sophisticated merge operations that happen on a routine basis (such as integrating international partner inputs into a plan for the International Space Station), automated merge and integrate capability can be developed.

The Plan Advisor

One of the fundamental design principles of the SPIFe toolkit is that the user's hand should not be forced by any integrated automated planning system. As a result, much of the feedback from the native constraint and resource engine as well as feedback from external systems is presented in a view called the Plan Advisor (Figure 3). The concept behind the Plan Advisor is that the human is in control of the plan, but he or she may selectively invoke help from automated systems.

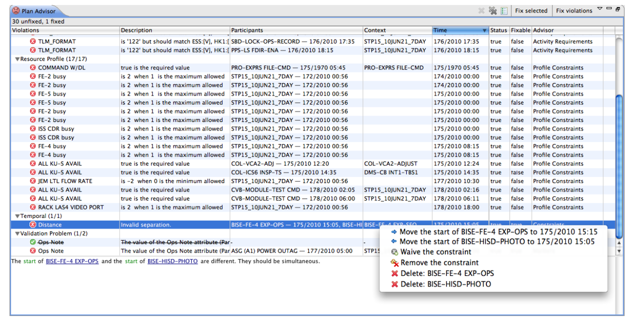

Figure 3: The SPIFe Plan Advisor

In most cases feedback is presented to the user in realtime after each plan edit. If a violation is determined to be fixable, either by native code or an external engine, users are presented with a context menu containing common fixes. If more extensive reasoning or search is required, users can invoke the capabilities of external systems via ”Fix Violations” commands which are also invoked from the Advisor.

In many cases violations are deemed acceptable, either due to a one-time exception, or more commonly an error or omission with the model or constraints themselves. In these cases, users can waive the violation and provide rationale. These waivers and rationale are persisted with plan data so an audit history is always preserved.

Resource Modeling

SPIFe has the capability to display resource usage effects that are derived from the schedule and visualize resource modeling information of varying kinds from coarse approximations to extremely high resolution simulation data. It also supports a multitude of higher fidelity simulation engines to display things like power, geometry (e.g. position of sun relative to spacecraft), or data usage. The results of external modeling tools are transferred seamlessly to the SPIFe toolkit for display in the context of the planning session: alongside the timeline, in columns in the table editor, in fields in an inspector pane associated with each activity, and in the Plan Advisor if necessary. This allow users to immediately see the effect of plan changes within the same context and debug issues that potentially result from them.

External Toolkit Integration

The implementation of SPIFe was designed from the ground up with integration of external planning and scheduling systems in mind. While some capabilities exist natively within SPIFe, most domains that have been encountered utilize high fidelity simulation and automated planning engines. Robotic Mars and International Space Station missions alike utilize agency and/or center wide standard modeling engines that simulate geometric, electrical and thermal fluctuations of the environment and physical hardware for presentation

Figure 4: The SPIFe timeline displaying several predict plots from a data model

to the planner. Planning engines have also been integrated that allow external systems to propose modifications to the currently editable plan. Such systems reconcile temporal and resource constraints and can be as complex as requiring distributed system architecture, or as simple as scripts that reduce planning redundancies within the product.

The integration with such systems has been made possible through client side translation of internal SPIFe models to external system models. These models are then communicated through RESTful interfaces, JNI, XML-RPC and/or the spawning of new processes to evaluate the plan and return simulated resource values and/or proposals for plan modifications which can then be applied by the user while maintaining the undo/redo functionality that allows users to back out changes if necessary to correct overlooked constraints and or resource allocations.

Each deployment of SPIFe has always come with unique challenges and thus unique engines and systems to integrate with. For the Phoenix Mars Lander, SPIFe integrated with the Europa planning engine to fix temporal violations, APcore which provided high-fidelity modeling of data acquisition and transfer, and the JPL-developed Multi-Mission Power Analysis Tool (MMPAT) for high-fidelity power and thermal modeling. For the International Space Station Power simulation product, SPIFe interfaces with numerous tools, bringing together numerous different modeling and simulation engines in a single, consistent user interface. These include the Spacecraft Electrical Equipment Database (SEED), Electrical Power Load Model (EPLM), the Battery and Solar Array Model (BSAM), Flight Dynamic Planning and Analysis (FDPA), Robotic Shadowing Calculator (RSC) and Solar Array Constraint Engine (SACE).

Integration with all of these high fidelity engines not only allow for the continued use of domain specific systems, but allow SPIFe to leverage years of usability testing to make the presentation of such capabilities intuitive, while not compromising on the computational requirements to run a safe and efficient mission.

Architectural Foundations

The modeling capability in SPIFe employs a widely utilized modeling framework called the Eclipse Modeling Framework (EMF). It is primarily an implementation of the Object Management Group's (OMG) Meta Object Facility (MOF). This capability allows SPIFe to tap into an extensive library of support services that support the generation of metamodels to describe domains using UML based tools, XML schemas, database tables, by hand using Eclipse based tooling, as well as a host of other techniques that continue to evolve.

This standardization increases flexibility while reducing the overhead that comes with use of planning and scheduling tools. Many deployments start with a careful analysis of the high level information planners wish to specify, and codify them into models. Many times, these specifications already exist in the form of database or XMl schemas, in which case the use of the EMF/ MOF capabilities make integration with existing systems trivial. If no such standards currently exist, industry standard tools can be utilized to create such models quickly.

Behavior of SPIFe planning models is specified through the use of various modeling languages as defied by the automated systems that SPIFe utilizes. In many instances, the native SPIFe capabilities are used which leverages JavaScript to take the metamodels and specify the effects, constraints and conflict resolution strategies once added to the plan. The use of JavaScript itself allows for the leveraging of a great deal of shared development resources in terms of tooling and documentation support.

In most cases however, the specific deployments of SPIFe in domains use external engines that typically have their own specific domain specific languages (DSL) to define the metamodels. In such cases, the EMF / MOF capabilities are only utilized to allow the UI to be configured for data entry and visualization of the information. The data is thus sent and returned to and from planning and scheduling engines asynchronously, keeping both the automated and manual panner in the loop at all times.

Concluding Remarks

Missions understandably set relatively high bars when it comes to stability, control, efficiency, and transparency in their operations processes. This may be especially true in missions with tight tactical planning cycles. Here, plans must be assembled quickly, and it must be widely understood why a plan has been assembled the way it has before a commitment is made to sequence it and execute it. The addition of automated planning technology then further accelerates the planning process. The focus on mixed initiative planning, where plan flaws are noted and repair assistance is provided, greatly contributes to transparency and control, without which rapid planning in a tactical operations context is far less useful. The ability to work with plans that are invalid from the perspective of the planning model allows users to incrementally build and repair a plan they understand and can explain.

In addition, experience suggests there will always be nuances, exceptions or changes to how a mission chooses to operate a spacecraft, and there isn't time during the tactical cycle to bring the existing planner model into agreement with the ground truth about the rover as understood by the operators. Having close control over the modifications the planning technology suggests for the plan is crucial in these situations.

Acknowledgments

None of the work presented in this demo would have become a reality without the dedication and passion of several remarkable teams. We would like to thank the JPL, JSC, and Ames development and design teams for their numerous contributions to the SPIFe toolkit.

References

Bresina J., Jonsson A., Morris P., and Rajan K. 2005a. Activity Planning for the Mars Exploration Rovers, Fourteenth International Conference on Automated Planning and Scheduling, Monterey, 2005, pp. 40-49.

Bresina J., Jonsson A., Morris P., Rajan K. 2005b. Mixed- Initiative Planning in MAPGEN: Capabilities and Shortcomings. Workshop on Mixed-Initiative Planning and Scheduling, ICAPS05, 2005.

Bresina J., Morris P. 2006. Mission Operations Planning: Beyond MAPGEN, Second IEEE International Conference On Space Mission Challenges for Information Technology (SMCIT), Pasadena, 2006.

Fox J. M., Norris J. S., Powell M. W., Rabe K. J., Shams, K. S. 2006. "Advances in Distributed Operations and Mission Activity Planning for Mars Surface Exploration", AIAA SpaceOps 2006, Rome, June 19-23, 2006.

Frank J., Jnsson A., 2003. Constraint-Based Interval and Attribute Planning. Journal of Constraints Special Issue on Constraints and Planning.

Maldague P., Ko A., Page D., and Starbird T., 1998. APGEN: A multi-mission semi-automated planning tool. First International NASA Workshop on Planning and Scheduling, Oxnard, CA, 1998.

McCurdy, M., Connors, C., Pyrzak, G., Kanefsky, B., and Vera, A. 2006. Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed- fidelity success. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Montral, Qubec, Canada, April 22 - 27, 2006).

Norris J. S., Powell M. W., Fox J. M., Rabe K. J., Shu I. 2005a. "Science Operations Interfaces for Mars Surface Exploration," 2005 IEEE Conference on Systems, Man, and Cybernetics, Big Island, HI., October 15, 2005.

Norris J. S., Powell M. W., Vona M. A., Backes, P. G., Wick, J. V. 2005b. "Mars Exploration Rover Operations with the Science Activity Planner," IEEE International Conference on Robotics and Automation, April 2005.

Tollinger, I., McCurdy, M., Vera, A. H., and Tollinger, P. 2004. Collaborative knowledge management supporting mars mission scientists. In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (Chicago, Illinois, USA, November 06 - 10, 2004).

Aghevli A, Bachmann A., Bresina J., Greene K., Kanefsky B., Kurien J., McCurdy M., Morris P., Pyrzak G., Ratterman C., Vera A., Wragg S., 2006. "Planning Applications for Three Mars Missions with Ensemble", AIAA SpaceOps 2006, California, June 2006.