Constraint and Flight Rule Management for Space Mission Operations

J. Barreiro*, J. Chachere*, J. Frank**, C. Bertels***, A. Crocker***

*Stinger Ghaffarian Technologies, Inc. NASA Ames Research Center, Moffett Field, CA, USA
e-mail: {javier.barreiro, john.m.chachere}@nasa.gov

** NASA Ames Research Center, Moffett Field, CA, USA
e-mail: jeremy.d.frank@nasa.gov

***NASA Johnson Space Center, Houston, TX, USA
e-mail: {christie.d.bertels,alan.r.crocker}@nasa.gov

Abstract

NASA's Mission Operations Directorate (MOD) has formalized thousands of operational constraints to help govern human spaceflight missions. MOD collects, develops, documents and applies these constraints to ensure the safety of the crew, as well as proper operation of the spacecraft systems and payloads. These constraints are stored in human-readable documents and also used to configure tools used by the flight controllers who plan and fly missions. We have begun developing a novel capability for authoring and maintaining such constraints called the Constraint and Flight Rule Management system (ConFRM). ConFRM provides history tracking and commenting features that allow users to trace the heritage of constraints throughout their lifecycle. ConFRM maintains links between related constraints, and between constraints and source information used to create the constraints. ConFRM uses these links to ensure consistency between constraints throughout their lifecycle, and provides authors and reviewers the ability to navigate between constraints and related data. Finally, ConFRM enables export of constraints into many different forms, including human readable documents and tool configurations, thereby eliminating manual labor and reducing transcription errors.

1 Introduction

Thousands of formalized operational constraints govern human spaceflight missions. NASA's Mission Operations Directorate (MOD) develops, documents and applies these constraints to ensure the safety of the crew, as well as proper operation of the spacecraft systems and payloads. During pre-flight planning, NASA and its partners systematically collect, develop, and document these constraints in operational products. The Program and vehicle vendor deliver many operational constraints to MOD. MOD develops other constraints as technical forums conducted with partners discuss and agree to mission details. This process provides the Flight Control Team (FCT) a set of approved operational documents with embedded constraints for use during mission execution. MOD provides these documents in paper form and online searchable databases. These constraint products include Flight Rules and Planning Ground Rules and Constraints. MOD also develops and maintains other documents that often include constraints, such as: flight procedures, systems briefs, operational handbooks, and mission timelines.

During training, flight controllers and related personnel learn all of the constraints relevant to their disciplines and configure a variety of tools to help enforce those constraints. During nominal operations, the FCT and crew continually monitor the spacecraft, crew, other systems, and the operating environment to determine whether or not the constraints remain satisfied. During off-nominal operations, the constraint documents indicate corrective actions the FCT should take in order to return to an acceptable mission state.

The following scenario illustrates a common life-cycle of operational constraints. Suppose hand-held radios onboard the space vehicle interfere with communications equipment used during spacewalks. One group in the Flight Control Team might write a constraint communicating that the crew should turn off these radios prior to initiating spacewalk operations, then might link this constraint to an external Hazard Report document that describes the potential for interference. Also, a second group in the Flight Control Team might write a constraint to ensure the crew's plan contains an explicit action to turn the radios off, then might link the two constraints. Over the course of several missions, there might be changes to the specific details of the constraint (such as the number of radios that must be turned off, the type of radio, and the crew procedure that details how the radios should be turned off). Also, these originally mission-specific constraints might become generic constraints that affect every flight. A typical constraint document appears in Figure 1.

B15-24 Amateur radio Inhibit for evas [HC] [RC]

AMATEUR RADIOS SHALL BE DEACTIVATED DURING ISS EVA'S. ®[042706-7156C]

Per Hazard Report RSCE-0015-INT (February 15, 2005), Non-ionizing (radio) irradiation of the ISS RS (Cause 2), the SPUTNIK-SM is required to be deactivated during EVA.

In addition to the hazard, EV/ESCG analyses indicate that amateur radio transmissions (SPUTNIK SM and FGB VHF) can degrade the Orlan Korona link (2-way simplex voice between the Orlan and SM) by up to 47.2 dB and the Tranzit-B link (provides telemetry from the Orlan to the SM) by up to 1.4 dB. Amateur radio operations can degrade the EMU SSCS links by up to 23.8 dB. To prevent the potential interference during the EVA, the crew will power off the ISS amateur radio equipment in both the SM and FGB.

FLIGHT/STAGE EFFECTIVITY: ULF1.1 AND SUBS ®[042706-7156C]

3.9.3 HAM Equipment power down

A 5 minute activity will be scheduled for the ISS crew to deactivate both the Ericsson (GE) VHF and Kenwood D-700 HAM radio units prior to Russian vehicle events and U.S./Russian EVAs.

Rationale: Safety of flight issue due to possible frequency interference.

Source: ISS MER Safety, FR B 15-24 (EVA)

Figure 1. Documented constraints on Ham radio operations.

Operating constraints may change from mission to mission. A constraint created for a specific flight may be deemed generally applicable. Or, variations in vehicle configuration may lead to different crew scheduling constraints. Similar operational constraints are developed independently by different organizations. Two constraints might contain identical information, yet today these documents might be maintained by different parts of the Flight Control Team. As a result, the documents may be mutually inconsistent because changes are not mirrored. Manual input of constraints into machine-readable formats (e.g., for automated planning, mission analysis, and mission monitoring tools) occurs after documenting the constraints. The resulting process is inefficient and error prone.

We have begun developing the Constraint and Flight Rule Management system (ConFRM), which is a novel capability for authoring and maintaining constraints. ConFRM provides history tracking, commenting, and notification features that help users to track and advance constraints through their lifecycles. ConFRM maintains links between related constraints, and between constraints and source information used to create the constraints. ConFRM uses these links to ensure consistency between constraints throughout their lifecycle, and provides users the ability to navigate easily between constraints and related data. Finally, ConFRM enables exporting constraints in different forms including human readable documents and data configured for partner applications. These features are intended to minimize labor, accelerate changes, and reduce errors in development and operations.

2 Previous Work

The AI planning community has previously addressed the creation and management of consistent operational constraints to drive automated planning. Existing tools can automatically detect ill-formed rules, detect mutually inconsistent rules, and infer rules from plans [2,3]. However, the task of managing these operational constraints for human spaceflight offers some unique challenges. The constraints must be documented so that both people and AI planners can use them. The constraints must be created by a large, distributed team of knowledge engineers. The constraints must be developed by experienced spaceflight operators and engineers who are not AI experts. While rules for automated planners have been extracted from documents e.g. for Orbital Express [7], this is not common practice today. Lastly, the gradual change of constraints over long periods of time introduces the problem of 'lifetime rule management'. The Procedure Integrated Development Environment (PRIDE) [8] [9] is a tool similar in spirit to ConFRM that is used to develop human spaceflight procedures. Like operating constraints, procedures have specific structure, embed references to commands and telemetry that may change over time, refer to other procedures, and exist in multiple forms. PRIDE offers many similar features to ConFRM, and ConFRM drew upon the experience of PRIDE development.

3 State of the Practice

The FCT documents several classes and properties of operational constraints and associated products. The FCT uses *Flight Rules (FRs)* to document the vehicle and programmatic constraints within which the FCT must operate and to outline decisions planned in advance to minimize the amount of realtime discussion. Mission planners (members of the FCT responsible mission timelines) use *Ground Rules and Constraints (GR&Cs)* and *Crew Scheduling Constraints (CSCs)* to help plan the mission's daily activities. *Workarounds* are a special class of constraint to address inconsistencies between a system's intended behavior and its true behavior. These constraints often arise because of software bugs, hardware anomalies, or design defects.

Within these broad categories are additional classes of constraints. For example, FRs may serve to define a system's state (e.g. nominal, lost, degraded) or whether a capability exists (e.g. able to rendezvous or land), while other FRs reference or detail these foundational rules.

In addition to having classes, constraints may have specific properties. Human spaceflight mission constraints evolve over time, as new systems are flown, older systems degrade, and experience is gained with existing systems. *Generic* constraints apply to all missions. *Flight-specific* constraints are specific to a mission's payload, objective, system configuration, or environment. For a typical six-month period, the ISS FCT manages 1000 generic FRs, 100 flight-specific FRs, 300 GR&Cs, and 100 CSCs. Some constraints apply throughout a mission; others apply during specific mission phases (e.g., when a Shuttle is docked). Some constraints apply to specific spacecraft (e.g., ISS or Shuttle); others apply to specific subsystems on a spacecraft (e.g., the ISS thermal control system). Constraints such as FRs may be marked with annotations that direct members of the FCT to perform specific actions. For example, the Flight Rule in Figure 1 has an [RC] annotation, which requires coordination or notification of Russia, one of NASA's International Partners, in the event the rule applies or gets updated.

Operational constraints are derived from one of several sources. The FCT uses documentation provided by the spacecraft contractor. For example, the *Operations Data Book* (ODB) serves as the authoritative source of the spacecraft performance characteristics, behavior, and certification limits. Many operational constraints are developed in order to mitigate a hazard documented in a *Hazard Report*. Many more are derived from engineering analysis of spacecraft systems, either performed by MOD or other organizations. Yet more operational constraints are derived from the operational experiences of FCT and crew, as documented during pre-flight technical forums, training sessions, or post- mission sessions. Constraints ensure traceability by referencing a document's title, number, author or date.

Constraints are prepared for a specific mission using a lifecycle approval process that becomes increasingly rigid as the mission date approaches. The constraints are initially written as documents in Microsoft Word. Constraints often require input or review from many members of the FCT, even though the constraint may pertain to only one physical subsystem of the spacecraft. The coordination process is performed by sharing of documents, by email, and by use of comments and change tracking features of Word. In order to become approved for use on the mission, constraints must pass through a formal configuration management process. This process ensures all stakeholders of the constraints' content and appearance have reviewed the rules; these stakeholders include FCT, Program/Project Management, Safety, Engineering, International Partners, and Book Managers. It also requires manually merging constraints documented in multiple versions, and ensuring linked constraints are consistent. Finally, it requires ensuring all tentative content in constraint has been finalized. Constraints are also organized into books or volumes, and published to a read-only location to ensure no untracked modifications take place prior to the mission. Books are organized by constraint type (e.g. FRs in one book, GR&Cs in another), spacecraft (e.g. ISS rules in one volume, Shuttle rules in a second, volume, joint rules in a third volume). Books and volumes are both printed and posted electronically.

The FCT uses a variety of software tools to plan and monitor missions in order to ensure compliance with the constraints or to determine when enforcing a constraint requires taking action. Tools fall into three classes. *Planning* tools are used to generate sequences of actions to be performed either by automated spacecraft systems (e.g. Timeliner, a system for onboard automation), the FCT (e.g. commanding thruster firings, solar array orientation or mode changes) or the crew. *Analysis* tools are used to evaluate plans or configurations in order to calculate resource quantities (e.g. available power or fluid levels) or properties of plans or configurations (e.g. equipment – bus interconnection tolerance to electrical system failures). *Monitoring* tools receive data in real time from the spacecraft and display it to flight controllers after a small amount of processing. Today, these tools are manually configured by the FCT as the constraints are approved. Members of the FCT trained in the use of these tools read the documented constraints then configure the tools appropriately. For example, a GR&C indicating that hand-held radios onboard must be powered off prior to ISS commanding events would require configuration of the planning system to flag this constraint. As another example, if a FR requires taking action if the temperature of the Shuttle during re-entry exceeds a specified value, a flight controller would have to look up the telemetry items or run analyses corresponding to temperature measurements of the Shuttle, and configure the monitoring systems to provide visual or audible (or both) cues to the flight controllers as the temperature approaches the limit. Preliminary versions of constraints may change prior to approval, so changes in constraints will prompt tool reconfiguration.

The current process of handling operational constraints is time consuming and prone to errors. The constraint management processes are particularly challenging when changes occur late in the process. These changes can occur either during training or flight for several reasons.

The constraint document approval process is long and labor intensive. Representatives from many internal and external organizations must review each document. Office products inadequately support this collaborative process. For example, much effort goes into enforcing format standards and reconciling multiple document versions.

Different parts of the FCT use operational constraints that share significant content. However, different groups of people author and manage those constraints. Inconsistencies between these constraints are often caught late in the process, introducing needless overhead and risk.

Constraint documents generally omit context, which help indicate how the document was created, its relation to other documents, and each stakeholder's complete rationale. Without that context, MOD personnel have difficulty evaluating the constraint when conditions change (such as when maintaining and upgrading the constraint documents).

It is difficult to search, determine applicability, and visualize these documents in training and operations so that their content can be understood and used as intended.

There is no support for automated links or data exchange with external applications. This gap delays flight controller access to many related tools (such as hardware description databases, planning systems, and monitoring tools). The inability to link and exchange data automatically leads to difficulty maintaining consistency among the many related constraints and applications.

4 ConFRM tool design

NASA has designed and prototyped a software solution called Constraint and Flight Rule Management (ConFRM). ConFRM's approach to authoring and managing operational constraints addresses the problems described above. The ConFRM design philosophy is to take the best features of text editors, markup, email, document version control, automated reasoning, and the World Wide Web that are suited to the task of authoring operations constraints and package them in a light-weight, easy to use form.

ConFRM allows authors to explicitly link a constraint to the many spacecraft command and telemetry descriptions, databases of hazards, previously created operational constraints, and analysis products that the constraint references. ConFRM can establish links to these products either manually or automatically. ConFRM can also detect changes to product content and location, so the constraints and links are always up-to-date.

ConFRM enables export of relevant information from operational constraints to planning, analysis and monitoring tools, thereby reducing the effort in mapping the documented constraint to the tools used to ensure the constraints are followed. ConFRM's approach to capturing constraint knowledge in a central database also allows each group to export the content it needs, reducing duplication of effort and operational constraint mismatches between groups.

The ConFRM prototype includes basic error and inconsistency detection supported by formal modeling. The NASA team is evaluating a promising enhancement to automatically integrate constraints with monitoring and planning software.

Figure 2 shows the three layers of ConFRM's technical architecture:

- Business Layer: encapsulates all of the business functionality that ConFRM provides, this includes constraint lifecycle, history version control, event notification, comment management, data search and retrieval, error checking, authentication and authorization. A plug-in mechanism allows existing modules to be enabled/disabled, and eventually will allow new modules to be loaded at run-time.
- Persistence Layer: provides persistency and retrieval mechanisms for all the data managed by ConFRM. A relational database is used to maintain the latest version for all the constraints, this allows ConFRM to scale well and to provide rich searching and reporting functionality. Historical data is maintained in a separate repository. The Storage Layer is not accessible directly outside of ConFRM, instead, it is encapsulated as part of the data and history services offered by the Business Layer.
- The Presentation Layer provides a rich authoring UI with intuitive formatting. An alternative, lightweight web UI can provide access for casual and external users (such as hardware manufacturers, whose role is limited to providing technical details for some constraints).

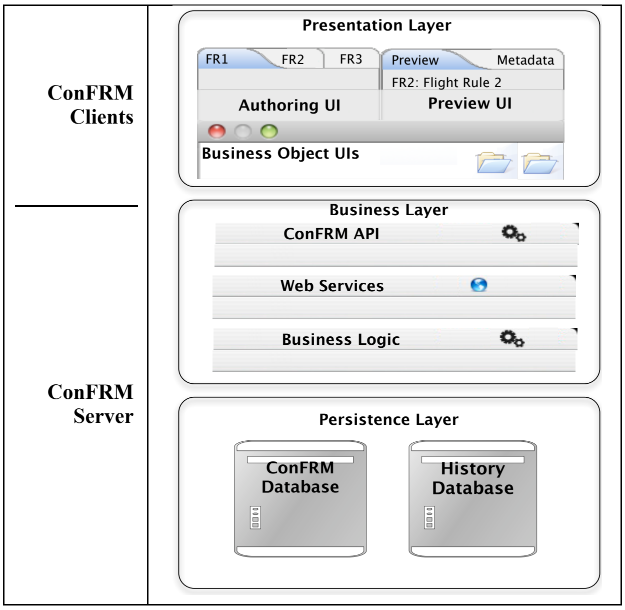

Figure 2. The ConFRM architecture includes a presentation layer that faces users, a business layer that hosts reasoning modules, and a persistence layer that stores constraints and related data.

All the services of the Business and Storage

Layers are encapsulated by the ConFRM Server, which can be accessed remotely by any client application that can provide the necessary credentials. The Presentation Layer for ConFRM interacts with the ConFRM server through its public API. Partner applications will interact with it in the same way for data exchange and business workflow execution in the future.

The principal business object created by users of ConFRM will be constraints, which are described further in the next section. However, ConFRM allows users to create other business objects associated with a constraint to facilitate authoring. ConFRM maintains a list of comments on a constraint to enable coordination and discussion of the contents of a constraint. ConFRM will allow users to maintain a dictionary of definitions and acronyms. ConFRM will create stored reports as a result of queries on constraints. ConFRM will facilitate permissions on constraints using a set of user and authorization objects. Finally, ConFRM will store a log of manipulations of a constraint, including archives.

Constraint Structure

All constraints have similar underlying structure. The *content* of a constraint is a description of the constraint, be it a directive or statement of a condition to maintain or avoid, and an explanation to clarify the constraint. The content of a constraint is organized in outline format. Each outline element is one or more paragraphs of plain text, Rationale, Table, Attachment, Image, Limit, or Model. Rationale is a justification for the constraint, and is typically more detailed than the constraint itself and is formatted differently than the constraint text. Tables may contain text or numerical quantities. Limits are typically bounds on a (set of) numerical quantities, and are designed to facilitate configuration of real-time telemetry monitoring applications. Models are formal descriptions of activities, and are designed to facilitate configuration of automated planning applications. Since limits may be unknown or tentative, ConFRM lets authors mark limits as To Be Determined (TBD).

In order to facilitate both the authoring process and duplicate existing rules, text can include Links, Pointers, and References. Internal Links designate a connection between two constraints or between a constraint and a ConFRM Business Object, and are navigable within ConFRM. External Links designate a connection between a constraint and a document available via a service connecting ConFRM to an outside data source such as a Web page. Pointers allow navigation within a single constraint. References are non-navigable document identifiers. Figure 3 shows a ConFRM interface for developing constraint content and links.

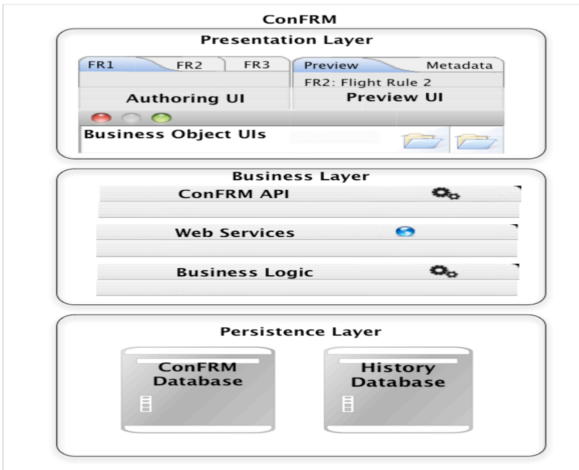

Figure 3. ConFRM distinguishes between constraint content (shown as text and data - see above) and constraint presentation (shown as traditional documents with links - see Figure 1).

In addition to content, all constraints contain *Metadata* that describes properties of the constraint. Metadata includes Name, ConFRMID, Number, Type, Status, Vehicle System Applicability, Program, Flight Applicability, Mission Phase, Annotations, Version, Create and Modified Dates, and Keywords. Constraint type controls needed content and appearance (e.g. Flight Rules need different content than Workstaiton Limits). The Status indicates its lifecycle maturity (Draft, Open, Approved, Retired). The ConFRMID ensures that a constraint has a unique identifier throughout its lifecycle, since its Name may change, and its Number may not be assigned until late in the lifecycle. Flight Applicability indicates the set of flights a constraint applies to (e.g. ISS Increment 20, Orion Pad Abort 1); if it applies to every flight it is considered Generic. Program indicates what program (e.g. ISS, Constellation, Joint operations). Annotations typically are coordination instructions to the FCT (e.g. a constraint may apply to both the U.S. segment and the Japanese segment of ISS, so a [J] indicates Japanese coordination). Vehicle System Applicability indicates what part of the spacecraft the constraint influences (e.g. power, thermal, communications). Finally, Mission Phase indicates when a constraint applies during a typical mission (e.g. only during Ascent or Docked Operations and EVAs).

Creating Content

ConFRM uses an Editor window to capture content, and a Preview window to show how a constraint will look when it is published. Since most text is organized as an outline, the Editor provides automatic outline features. Each type of content can be inserted into a rule either by typing, using a keystroke accelerator, or by using a button on the Editor. A style sheet controls the mapping of content to the Preview window (e.g., Flight Rules have capitalized body text and italicized rationale).

The Editor window uses structure detection to automatically identify internal and external links. Authors can use either ConFRMID, Name or Number to specify internal links. External links are specified as URLs. ConFRM supports creation of links to constraints not yet created and changing of link properties. An interesting nuance of this is that the user must be able to both access the properties pane to change a constraint property and navigate to the destination; this is supported using different mouse gestures. The editor highlights changes in a constraint and provides visual indicators in a tab at the top of the tool to inform users if a constraint has been edited and needs to be saved.

Many constraints (especially Workstation Limits) will consist of tables of telemetry references, accompanied by limits indicating that a system is nominal (e.g. green), off nominal (yellow) or outside acceptable limits (e.g. red). To enable this, ConFRM allows authors to select a set of telemetry reference business objects, drag them into a table, then specify the relevant limits. Figure 4 illustrates the management of Workstation Limit tables using ConFRM.

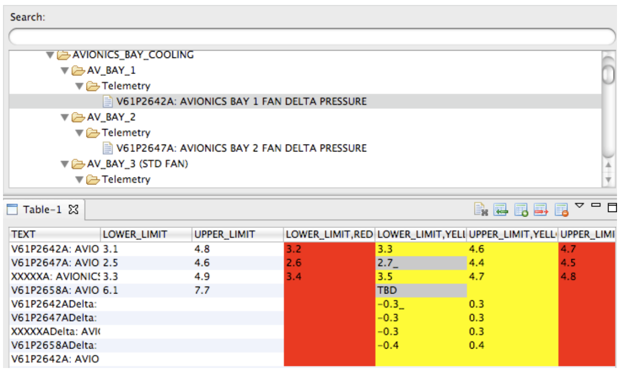

Figure 4. ConFRM represents ‘Workstation Limits’ – during operations, telemetry items (selected in upper pane) should not exceed these values (lower pane).

Business Objects

ConFRM organizes content created by authors and content from external applications into *Business Objects*. These are presented to authors to help them create constraints efficiently.

ConFRM will subscribe to descriptions of the spacecraft subsystem architecture, and associated commands and telemetry. This information will be presented to authors as a System Representation business object that allows authors to look up system identifiers, associated telemetry, and easily include them in a constraint [reference to PRIDE paper or papers]. ConFRM will also subscribe to Operations Data Books provided by spacecraft and subsystem contractors that describe operating limits that influence constraint development. Other data includes standard operating procedures, programmatic milestones, analysis documents, and meeting notes.

Authors specify Metadata in a separate pane. Most of the metadata fields are discrete options, and can be specified using checkboxes. Metadata is prominently displayed as a header in both the Preview and Editor windows.

History, Commenting and Version Control

ConFRM provides integrated features to manage versions of constraints and support maturation of a constraint through its lifecycle. Broadly, a constraint’s lifecycle evolves as follows: Draft constraints are either brand new constraints or copies of constraints from a completed mission that must be changed in preparation for a new mission. One or a small number of people usually creates draft constraints. Open constraints are mature enough for a broader review, but not ready for use in operations. Approved constraints are ready for use in operations, and under configuration management; while they can be changed it requires a formal Change Request process to do so. Retired constraints are archived for reference but not approved for use in operations.

Authors can save constraints without creating a new version, or create a new version of a constraint that will be stored in an archive. ConFRM will suggest a name for a new version that the author can override. Creating new versions does not influence lifecycle status. Changing lifecycle status always results in a new version. Metadata changes are stored in the current version automatically, while content changes require the user to explicitly save or create a new version.

ConFRM can show the version history of a constraint. After creating a new version, ConFRM logs the date, time, and user along with any comments. Users can also create more versions any time they wish. Also, ConFRM automatically saves constraint versions at key times in the constraint development. Every lifecycle change will automatically create a new version. Before any new author begins editing a constraint, ConFRM saves a version. Metadata changes and comments do not lead to new saved versions. Users can view any stored version.

Users can create comment threads to discuss a constraint. As described in the description of Business Objects, these are not part of the constraint but are stored with the constraint. Comments are stored in threads presented as drill-down trees, and can also be sorted by name, author, create date. ConFRM also supports special types of comments to indicate requests for changing a constraint, and the disposition of those requests. Comments are stored with versions.

It is common to create new constraints from old ones. ConFRM allows any constraint to be copied in full as a new Draft constraint. ConFRM provides users the ability to compare two constraints, either from the version history or two current constraints. The differences are indicated visually in a separate UI pane.

Views and Search

ConFRM provides numerous search features to provide authors with the ability to find constraints or other business objects with specific content. The Constraint Explorer, System Representation, and Dictionary all provide a Search pane that restricts the list of viewed items. For example, the list of constraints can be restricted to only those with EVA in the Name. An additional Search menu item will search the current version of constraints to find those containing a text string. This search feature can also restrict the search to constraints with a particular status.

Business Logic

ConFRM's Business Logic provides several mechanisms to ensure that constraints have the needed content and are consistent with other constraints.

The ConFRM UI ensures that authors of a constraint provide minimum identifying information for a constraint at the time it is created (e.g. a name). ConFRM automatically assigns a ConFRMId. The Metadata pane uses checkboxes to ensure that only valid metadata is provided, which reduces errors. Lifecycle status advances are also controlled via the Metadata pane.

Links provide a key means of automating the detection of changes. Changes in command and telemetry specifications that influence the System Representation in ConFRM will result automatically in a list of affected constraints. Changes in a constraint will automatically result in a list of linked constraints that may also need to be changed. Finally, changes in Dictionaries will result in a list of impacted constraints containing the resulting dictionary definitions or acronyms that must be updated.

Business logic for permissions and authorizations are defined by the author of a rule, or the ConFRM administrator. These permissions govern who can edit a constraint, who can comment on a constraint, and who is notified when constraints change.

ConFRM can suggest internal links between constraints upon creation or modification. It is common to either reuse a flight specific constraint, create a generic constraint from a flight specific one, or vice versa. ConFRM can suggest creating a link between a newly created constraint and its ancestor constraint. ConFRM can also suggest linking two constraints that share links to the same ConFRM business object, such as a System Representation object or Dictionary entry.

ConFRM provides authors the ability to detect a variety of problems with constraints prior to publication. For example, ConFRM will detect are any TBDs in a constraint and report those to the author. ConFRM will also detect any links from a constraint to a constraint or other ConFRM business object that has not been created. ConFRM will allow users to determine whether constraints that apply in the same mission phase are inconsistent by analyzing the applicability metadata. If the content of the constraint is represented formally (i.e. limits or activity models) mathematical or logical reasoning can be used to determine inconsistency, empty bounds, uncovered cases or other possible errors.

Task Management and Notification

ConFRM aids the constraint development process by integrating email notification with key activities. For example, ConFRM automatically sends an email notice to a constraint's owner each time another user comments on it. ConFRM allows users to subscribe to a constraint, so that they will receive notices whenever there are changes or comments regarding that constraint. ConFRM sends an email reminder to users when deadlines are approaching for them. ConFRM also allows users to develop custom messages and send them to other users. These notifications help to drive progress on constraint development and ensure that relevant conversations occur and are tracked within the constraint context that ConFRM manages. Figure 5 shows the ConFRM authoring and commenting interface.

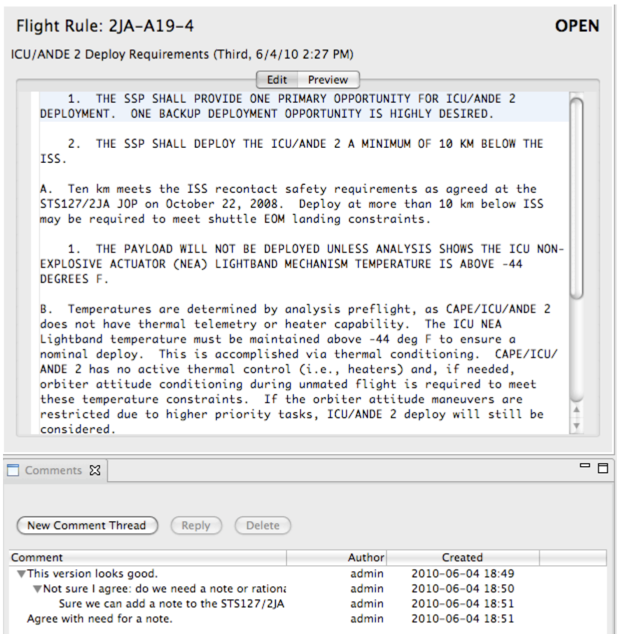

Figure 5. ConFRM supports collaboration in constraint authorship and review using comment threads linked to users via email.

ConFRM also manages a list of tasks that are pending action for each user. Users can view the work that remains before a constraint can be approved (e.g., removal of broken links, definition of metadata, and approval by different users). Users can opt to receive automatic email notification when their work items are due, or receive notification when their subordinates' work items are overdue. Providing the to-do list helps users focus on their deliverables and use the application efficiently.

The to-do and notification systems help support ConFRM's integration with a wide variety of related applications and document repositories. Hazard reports provide one example cited in the flight rule above. Hazard reports are managed outside ConFRM, but relate to constraints in many ways. ConFRM tracks which constraints are related to which hazard reports so that these related databases can be kept consistent continuously and easily.

Export

Constraints created in ConFRM will be exported to a number of different 'end-use' products. Constraints can be exported in a variety of formats for use by people, including documents and HTML. The style sheet governs font, color, layout and suppression of content that is primarily used for authoring. Workstation limits can be exported to configure telemetry monitoring applications such as Mission Control Technologies [8]. GR&Cs with accompanying formal activity models can be exported both to a format used by people, and also to configure a planning system (e.g. the Consolidated Planning System [10]).

5 ConFRM Implementation

ConFRM is implemented using numerous publicly available technologies. The Business and Persistence layers use Spring, a lightweight container that provides robust and scalable ways to provide a plug-in architecture, support industrial-strength authorization and authentication, and make the ConFRM Server API available remotely, in this case using Java Remote Method Invocation. The Persistence Layer is complemented by a MySQL database and Hibernate is used to automatically map the database into a Java class. The history of a constraint is stored in a Subversion repository. The Presentation layer uses the Eclipse Rich Client Platform. In addition to using many publicly available plugins, including text editors, the ConFRM team has also created special-purpose plug-ins for parsing, validating and presenting Constraint content is ways that are amenable to the ConFRM users.

6 Conclusions and Future Work

We have described ConFRM, a tool to help human spaceflight operators develop and manage constraints. ConFRM will integrate disparate data sources as well as historically developed constraints into a single repository that allows users to easily trace the history of constraints, edit constraints, link constraints to each other and other information sources, automatically check constraints for various problems, and publish constraints as readable documents or automatically export configuration information to other spaceflight operations tools.

While each feature of ConFRM has been prototyped, the individual features have yet to be integrated in the application. Partner applications and web services are still under development. We also expect numerous design revisions between delivery of the preliminary version of the system (in September 2010) and subsequent versions.

References

[1] Frank, J., "When Plans are Executed by Mice and Men." Proceedings of the IEEE Aerospace Conference, IEEE, Big Sky, MT, 2010.

[2] Simpson, R. M. Kitchin D. E. and McCluskey T. L. Planning domain definition using GIPO. The Knowledge Engineering Review. Volume 22 , Issue 2 (June 2007) pp. 117-134, 2007

[3] T. S. Vaquero ; V. M. C. Romero;, F. Tonidanel; J. R. Silva. itSIMPLE 2.0: An Integrated Tool for Designing Planning Domains. Proceedings of the International Conference on Automated Planning & Scheduling 2007, Providence, Rhode Island. Menlo Park, California, USA, 2007.

[4] Aghevli, A., Bachmann, A., Bresina, J.L., Greene, J., Kanefsky, R., Kurien, J.,McCurdy, M., Morris, P.H., Pyrzak, G.,Ratterman, C., Vera, A., Wragg. S., "Planning Applications for Three Mars Missions with Ensemble." 5th International Workshop on Planning and Scheduling for Space. Baltimore, MD, 2007

[5] Simon, G. Shaya, E. Rice, K. Cooper, S. Dunham, J. Champion, J. "XTCE: A Standard XML-Schema for Describing Mission Operations Databases," IEEE Aerospace Conference, IEEE, Big Sky, MT, 2004

[6] Izygon, M., Kortenkamp, D., Molin, A., "A Procedure Integrated Development Environment for Future Spacecraft and Habitats," Space Technology and Applications International Forum, Albuquerque, NM, 2008.

[7] Knight, R., Chouinard, C., Jones, G., Tran, D. "Planning and Scheduling Challenges for Orbital Express." Proceedings of the 6th International Workshop on Planning and Scheduling for Space, 2009.

[8] Trimble, J., Crocker, A. Reinventing User Applications for Mission Control. Proceedings of the 2010 Space Ops Conference, Huntsville, AL., 2010.

[9] Bonasso, R.P. Boddy, M., Eliciting Planning Information from Subject Matter Experts. Proceedings of the Workshop for Knowledge Engineering of Planning and Scheduling Systems, Toronto, CA., 2010.

[10] Frank, J., Morris, P.H., Greene, J., Hall, T., "The Challenge of Evolving Mission Operations Tools for Manned Spaceflight," 9th *International Symposium on Artificial Intelligence, Robotics, and Automation for Space*, Los Angeles, CA, 2008.