



Automated Derivation of Complex System Constraints from User Requirements

Kim Muery
NASA, Marshall Space Flight
Center
kim.muery@nasa.gov

Mark Foshee
Computer Sciences Corporation
mark.foshee@ums.msfc.nasa.gov

Angela Marsh
NASA, Marshall Space Flight
Center
angela.marsh@nasa.gov

Abstract

International Space Station (ISS) payload developers submit their payload science requirements for the development of on-board execution timelines. The ISS systems required to execute the payload science operations must be represented as constraints for the execution timeline. Payload developers use a software application, User Requirements Collection (URC), to submit their requirements by selecting a simplified representation of ISS system constraints. To fully represent the complex ISS systems, the constraints require a level of detail that is beyond the insight of the payload developer. To provide the complex representation of the ISS system constraints, HOSC operations personnel, specifically the Payload Activity Requirements Coordinators (PARC), manually translate the payload developers' simplified constraints into detailed ISS system constraints used for scheduling the payload activities in the Consolidated Planning System (CPS). This paper describes the implementation for a software application, User Requirements Integration (URI), developed to automate the manual ISS constraint translation process.



1. Introduction

The process of converting payload developer science requirements from a simplified representation in User Requirements Collection (URC) to a more complex representation required in the Consolidated Planning System (CPS) is a time-consuming and tedious process. The goal of the User Requirements Integration (URI) is to provide a software solution to replace the manual process of converting payload science requirements into CPS scheduling requirements. This goal is accomplished by providing

the PARCs a way to input the information that is referenced during the manual conversion process and by applying this information during an automated export to the CPS. This information includes ISS systems constraint modeling data, ISS rack power and data connections, ISS payload rack locations and configurations, trained crew assignments and CPS data required for scheduling.

2. Terminology

The following definitions are provided to assist the reader in understanding the Operations Concept presented in this paper.

2.1. Activities and sequences

Activities define the required duration, location, procedures and constraints for performing the activity. Sequences define the execution windows and the timing relationships between the members of the sequence. The sequence members may be activities or other sequences.

2.2. Timeline

A timeline is the result of scheduling the activities and sequences in a manner which results in a plan for conflict-free execution of required events. In a timeline, each activity and sequence is assigned a fixed start time and stop time.

2.3. Constraints

The term constraint is used to represent both resources and conditions. Resources, such as power, have an availability of some amount over time. Conditions have availabilities defined in binary terms and may be used concurrently by an unlimited number of activities. An example of a condition is the

availability of the Tracking and Data Relay Satellite (TDRS). Two types of constraints are referenced in this paper, URC constraints and CPS constraints. URC constraints are defined by the PARC in URI, for selection in URC by the payload developer while building his activities. CPS constraints are defined by the planning community in the CPS and are used during timeline development.

2.4. Constraint mappings

Constraint mappings are relationships that are defined between the URC constraint and the CPS constraint. CPS constraints are used to model and schedule the ISS systems usage in the CPS. The URI provides an import function for CPS constraints. Payload planners then use URI to develop constraint mappings to map the constraints available to the payload developer in URC to the constraints defined in the CPS. Constraint mappings are defined as either generic, i.e., applicable to any topology location, or location specific. Location specific relationships are used to assign CPS constraints that vary based upon topology location. Constraint mappings are organized in a tree structure as shown in Figure 1. This example shows the URC constraint UIP Power/Thermal mapped to many CPS constraints. The CPS constraints are listed under the Name node.

assigned a payload alias. Many activities and sequences may share the same payload alias.

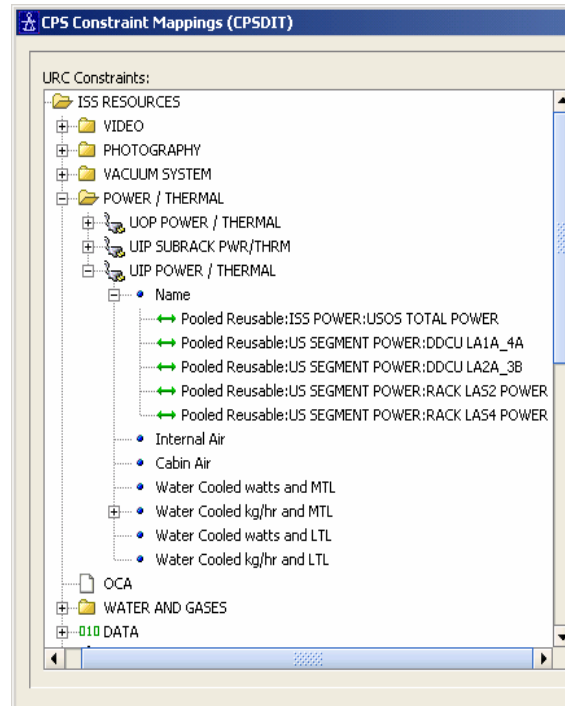


Figure 1. URI constraint mapping

2.5. Topologies

A URI topology is used to represent the ISS payload rack locations for the purpose of assigning location specific ISS constraints. These ISS constraints have been mapped to URC constraints with a location specific relationship. Topologies in URI are defined by creating segments within the topology and assigning locations to those segments. To each location, payload aliases and location specific constraint mappings are assigned. Figure 2 shows an example topology with a United States On-orbit Segment (USOS) containing four locations, LAS2, LAP4, LAO2, and LAO1.

2.6. Payload aliases

Payload aliases are payload identifiers used in lieu of the actual payload name because they represent not an actual payload per se, but a group of activities of the same payload that share the same resources due to their common location in the ISS. A payload alias may also be created to provide for the case where activities are of the same payload; but, they have different ISS crew trained for the activities. Each activity and sequence is

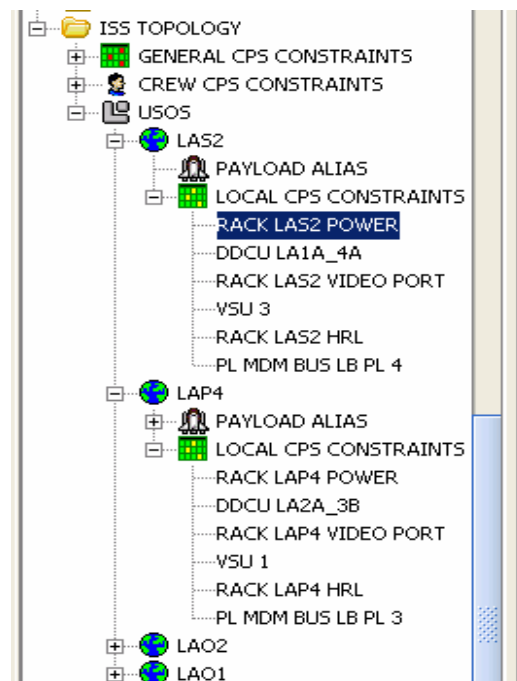


Figure 2. URI topology

2.7. Increments

Increments are operational time periods defined by the beginning and ending of a crew rotation. When new crew members begin to operate the ISS, a new increment begins and the prior increment ends.

2.8. Crew assignments

Crew assignments are defined in URI to identify the crew members that are trained on a particular payload. Crew assignments are made in URI to the payload alias, rather than the payload, to provide for the case where different activities of the same payload have different trained crew members. The ISS crew that is trained to perform a given payload is known in advance of planning the payload operations. The source of this information is not the payload developer. The payload developer selects only the quantity of crew members required to perform the activity. The actual crew resource assignment is made during the URI export process.

2.9. CPS attributes

CPS attributes are defined as descriptive items used in the CPS to sort, group and display the activities and sequences associated with a given timeline. These attributes include values for filter categories, class, origination and other descriptive information. CPS attributes are not a part of describing payload science requirements and therefore are not requested from the payload developer. The CPS attributes are added to the activities and sequences during the URI export process.

3. Operations concept

The payload developers use the URC to develop and submit their payload science requirements in the form of activities and sequences. Choosing from a pre-defined picklist, the payload developer builds an activity by selecting the required constraints for the activity. To assist the payload developer in defining his science requirements, ISS system constraints are presented for selection in URC at a basic level, such as, power, crew and water. Once the desired activities are defined, the payload developer builds sequences to link the related activities and promotes the sequences to the PARCs for review. Since the planning process is not the subject of this paper, actions that occur to review and accept the submitted sequences will not be discussed.

The URI provides the payload planner the capability to model an ISS topology. Each increment must have a topology assigned to it. ISS topologies change over time based upon new payloads being integrated, other payloads being terminated and by reconfigurations. The URI allows many topologies to be created. Topologies are defined by creating ISS segments and assigning locations to these segments. Examples of ISS segments are the United States On-orbit Segment (USOS) and the Japanese Experiment Module. Mapped constraints, that vary based upon location within the topology, are assigned to the locations within the topology. For example, power is allocated at a certain amount per payload rack. So, power must be tracked per rack, as well as, the total power usage for the ISS. Thus, when a payload developer requires power for his activity, the location of the activity is used to determine which rack power will be used during the activity. Figure 2 provides an example of the location specific constraint, Rack LAS2 Power, being assigned to the topology location of LAS2.

Constraint mappings are defined per topology. Constraint mappings link URC constraints to CPS constraints. Because there are embedded constraints within one URC constraint dialog type, CPS constraints are mapped to attributes of a constraint dialog. Section 4 of this paper provides further detail for constraint mapping to URC constraint dialog attributes. Constraint mappings also include a capability to define the desired quantity of the CPS constraint. The user may define the rate and constant for a linear equation to be applied to the URC constraint quantity. This capability is useful in cases where the desired CPS constraint has different units of measure than the URC constraint and where the desired CPS constraint is a fixed quantity. Constraint mappings may be defined as generic or location specific. Generic constraint mappings are applied to the activity regardless of the location of the activity. Location specific constraint mappings must be resolved using the location of the activity and the CPS constraints that are assigned to that location.

Crew assignments are defined in URI by the PARC per payload alias. During the URI export process, the payload alias for the activity being processed is referenced and the ISS crew assigned to the payload alias is applied to the activity according to the quantity requested. For example, if one crew member is requested for the activity, the crew member designated as prime for the payload alias is selected for the activity during export.

CPS attribute values are determined by the entire ISS planning community. Once these values are

defined, they must be used consistently by all planning participants to aid in producing the execution timeline. URI provides a capability for the payload planners to define the CPS attributes to be used for a given increment.

The export to CPS from URI is the automated portion of this process and is intended to replace the manual process of converting URC activities and sequences into CPS activities and sequences used to build an execution timeline. The goal of export is to translate the URC activities and sequences requested by the payload developer into activities and sequences that may be used directly in the CPS.

After selecting the desired sequences to export, the PARC selects the increment to be used. During the export process, the URI application replaces the URC constraints, selected by the payload developer, with the CPS constraints, according to the selected increment's topology and its associated constraint mappings. All CPS attributes for the selected increment are provided for the activities and sequences during the export process, as well as crew assignments for activities. Prior to the export to the CPS, activities and sequences are checked for completeness and required fields.

4. Constraint mapping implementation

URC constraint dialogs are tailored for the particular category of constraint. Figure 3 is an example of a data category constraint. For each desired data constraint, the payload developer is required to supply the data rate, data path (High, Medium, or Low), the data destination (Ground and/or Payload), data format (Bitstream, Packet, Bitpacket) and the downlink requirements (Near-real time or Real-time).

Several of the data constraint dialog attributes correspond to CPS constraints required for scheduling. Figure 4 illustrates the CPS constraints required to model a URC data constraint requesting the High data path. Based upon the payload alias location for the activity, one of the "Rack LAXx HRL" constraints and one of the "EXP Rx High Dig Tot" constraints will be added to the activity during export, as well as the HRDL Total, HRFM Digital Total and Digital Total. The URC activity contains one data constraint; the resulting CPS activity contains five data constraints. Values for the data rate, data destination, data format and downlink requirements will also be added to the activity. If the URC activity had selected the Medium data path, an alternate set of CPS constraints would have been applied at export. Refer to Figure 4 for a list of the CPS constraints required for the Medium Data Path.

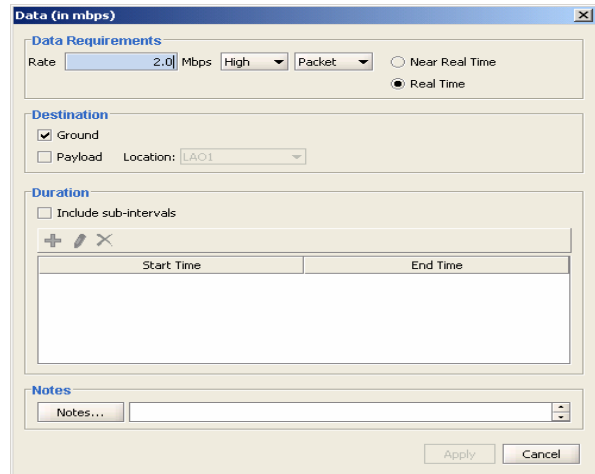


Figure 3. URC data constraint dialog

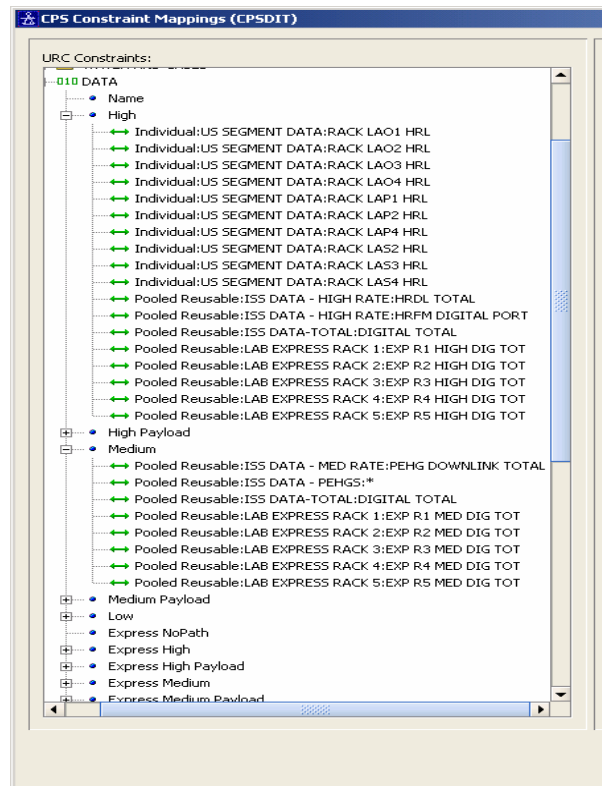


Figure 4. URI constraint mappings for data

5. Payload Planning System (PPS)

PPS is a suite of remotely accessible applications used for planning payload operations for the International Space Station. The main functions required for planning ISS payload operations are defining user requirements, managing the planning

lifecycle of the user requirements, scheduling payload operations, and scheduling data routes and configurations. URC, URI, CPS, and the Data System Routing and Configuration (DSRC) are the major software applications which comprise PPS and are used to perform these functions. Figure 5 depicts a high level view of the PPS architecture.

Payload developers use URC to define and submit their planning requirements. URC allows the payload developer to define their requirements by creating building blocks of activities and sequences to represent their payload operations and temporally linking them together on a drawing canvas. These activities and sequences are stored and organized in a centralized PPS database in such a manner to allow multiple payload developers and planners (with granted privileges) to share access to all activities and sequences associated with a particular payload. After defining their requirements, payload developers submit their sequences for review using the promotion feature of URC.

While the definition and promotion of user requirements are a part of the planning lifecycle, the major portion of this function is performed by planners and PARCs using the URI application. The initial step of the lifecycle is to perform increment setup tasks such as defining the URI constraints that payload developers use to define their requirements, creating increment and increment-generic attributes, creating topologies, importing CPS constraints, and mapping the URC constraints to the CPS constraints. Planners also use URI to assist payload developers in defining their requirements. Once payload developers submit their sequences, planners use URI to move sequences defined within URC through the lifecycle process. URI also has the capability to create and edit activities and sequences. This feature is used by PARCs to ready sequences for export.

Export is the URI function that automates the constraint derivation process for the activities submitted by the payload developer. It interfaces with CPS to export URC sequences and activities into the CPS database. Export transforms these sequences and their activities into the timeline import/export file format as defined by the Multilateral Distributed Planning Interface Specification (MuDPIS) [1]. Export then transparently initiates the CPS Loader application to load the sequences and activities into the CPS database.

CPS provides the functions necessary to model ISS constraints and their availabilities, as well as activities and sequences. CPS contains the scheduling engine used to build a timeline by assigning start and

end times for each scheduled activity while book-keeping the constraint usage and availability.

DSRC, although not germane to this topic, is discussed here because it utilizes products of the automated constraint derivation process. DSRC uses the timeline produced by CPS, including its activities and constraint availabilities. DSRC provides the capability to model the onboard data systems resources and scheduling constraints, and schedules the utilization of data system elements to produce a data flow plan (DFP). The DFP represents a schedule of data activities and the data configuration changes required for the onboard systems.

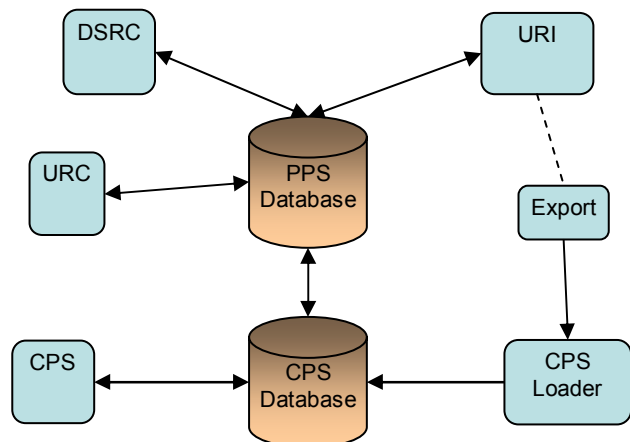


Figure 5. PPS architecture

6. Summary

Together, URC and URI provide a comprehensive solution to modeling payload science requirements. Using URC, the payload developer is shielded from the complexity of modeling the ISS system constraints, yet is able to completely specify the payload's requirements. Using URI, the PARCs are able to provide sufficient detail such that payload planners can accurately model and schedule ISS system usage in CPS. Finally, using the export feature of URI ensures a consistent, reliable approach to transforming URC constraints into CPS constraints.

The Payload Planning System (PPS) version 6.0 is currently supporting ISS for Increments 14 and later. PPS supports payload operations for the ISS user community, payload developers and cadre at the Huntsville Operations Support Center (HOSC) of the Marshall Space Flight Center. Although PPS, specifically the URC and URI components, has been presented using the ISS operations concept, the PPS could be tailored to fit a new need. The PPS is currently being considered to support the NASA Constellation program.

7. References

- [1] National Aeronautics and Space Administration, Multilateral Distributed Planning Interface Specification, Consolidated Planning System Cycle 19 Baseline, August 2005, SSP-50401-C19.
- [2] National Aeronautics and Space Administration, Payload Planning System (PPS) Software Requirements Specification (SRS) for the User Requirements Collection (URC) CSCI Revision 3, HOSC-SRS-667, December 2005.