

Requirements for the APT GUI

Steve Lubow

Original December 2000

Updated Feb 14, 2001

Updated March 5, 2001

1. Background

During the past several months a subgroup of the APT user group has been meeting frequently to discuss the user interface that will control the overall proposal development process. This document describes the requirements that have been obtained to date as a result of those discussions. The minutes of the APT user group meetings contain more details about various discussions that led to these requirements.

2. High Level Description of GUI Capabilities

a. Getting Started

- Upon entering the APT, users select the Phase of the proposal that they wish to develop. This will determine which tools are available. Users can then switch between Phase 1 and Phase 2 development while in the APT.
- Users will be able to import data from current or previous Phase 2 proposals.
- Online help will contain the APT documentation.
- Online directed examples will be available that display the sequence of steps required to carry out some common tasks.

b. Navigation

- Users will be able to work on multiple proposals of a given type (Phase 1 or Phase 2), while working within the APT. This allows information to be copied between proposals.
- Users will navigate through proposals by means of a hierarchical view of their proposals (similar to the hierarchical file chooser). This view lists the proposals and the items within proposals: targets, patterns, visits, and exposures.

- Users can add, change or remove proposal items within the hierarchical view. A simple click will provide a selection of operations (delete, copy, edit, move) as well as a choice of tools that can be run to view or change the selected item.

- Users can select items in the hierarchical view in order to provide a context (set of input properties) for tools. The detailed properties of the selected items will be displayed.

c. Interactive Tools

- Some of the tools will be interactive and visual, while others will operate in a batch mode. The interactive tools generally operate on proposal items, while the batch tools operate on the entire proposal (or at least an entire visit).

- Spreadsheet editors of the proposal provide a complete tabular view of the proposal (i.e. the data that will appear in the submitted proposal). In addition, they allow users to manipulate the view and change values. There will be spreadsheet editors for targets, visits, exposures, and patterns.

- Visual Target Tuner allows users to view and manipulate instrument apertures against an image of the target. Within this tool, the following two tools can be accessed.

- Archival Research Tool allows users to display information about observations from the HST data archive and perform duplication checks while in the Visual Target Tuner.

- Bright Object Checker determines whether there are bright object concerns for an exposure related to either instrument health and safety or science problems.

- Exposure Time Calculator determines the exposure time for a selected exposure or determines the S/N expected for a particular exposure time.

- Orbit Planner determines the layout of exposures in an orbit (Qwik-Trans), displays the results graphically, and allows for graphical tweaking of the orbit by operations such as copying, lengthening, moving, and deleting exposures.

- Visit Planner provides a graphical specification of visit dependencies (such as Visit 1 After Visit 2), and then displays possible scheduling times, as determined by Spike.

d. Working on a proposal item or template

- Users can select proposal items (targets, patterns, visits, and exposures) or templates to work on before or after entering interactive tools.
- Users can select one or more proposal item(s) to explore before entering an interactive tool. The selected proposal item(s) are then loaded into the tool.
- Users can change context (which proposal item is being viewed) while working within a tool.
- Users can enter a tool without a context. A user can later select a proposal item to store the properties of a tool object. If no appropriate exists, then the user can create a new proposal item and select it.
- Users will also be able to save information they have generated in a tool to a template for possible later use within a proposal. Templates do not appear directly in the proposal, but instead provide a library of information that can be applied to the proposal.

e. Working in GUI

- Within tools, users directly modify properties of proposal items in memory. Users write proposal and template data to disk through a Save operation in the File main menu.
- Only one tool can be displayed at any time. Tools will be able to display information from multiple proposal items (e.g., multiple exposures) and templates simultaneously within a single tool, where permitted by the tool.
- Users will be able to retract recent changes through the use of an undo capability.
- Changing a proposal item within a tool will not result in the automatic recalculation by the other tools. (For example, the user must choose to bring up the ETC to see the effect of an aperture change in the VTT on the exposure time. More seriously, a change in instrument in the VTT will likely leave an improper choice of filter in the ETC.)

f. Batch Checking and Submitting

- Users must run a batch checker before submitting a proposal. This

tool will carry out a series of proposal checks and report problems to the user, similar to the current processing in RPS2.

- The tool generates a graphical report on the proposal, like the current Description Generator in RPS2.
- If problems are found, users will re-enter the appropriate tools to fix problems.
- The batch tool will allow the user to select a subset of the checks, similar to current processing in RPS2.
- These checks must be satisfied before a user can submit the proposal.
- The checks run by the batch checker include:
 - consistency across proposal items (were all the targets used?)
 - feasibility
 - schedulability
 - guide star availability
 - bright object checks
 - This is only a required check for cases where health and safety issues are involved (e.g., STIS MAMAs).
- Proposal Submission Tool submits a proposal to STScI. It checks that the global checker was successfully run before allowing submission to proceed.

3. Requirements for Initial Screen

R3.1 When the APT begins, there shall be an initial screen with some general descriptive information about the APT.

R3.2 The initial screen shall allow a user to elect to enter Phase 1 or Phase 2 proposal development

4. Requirements for Phases 1 and 2

Note: In discussions below, two designs for the hierarchical editor (HE) are considered. In the HE Tool (HET) design, the HE runs as an APT tool. In the HE Mode (HEM) design, the HE runs within the hierarchical viewer.

Both Phase 1 and Phase 2 screens will be similar in appearance. Below we describe those requirements for both phases.

a. Mode Chooser

R4a.1 There shall be a mode chooser to allow users to switch between Phase 1 and Phase 2 of proposal development.

R4a.2 Users shall be able to change modes at any time.

R4a.3 The state of the proposal development shall be retained when a user switches between modes.

b. Main Menu Bar

R4b.1 There shall be a main menu bar that contains the usual PC-like capabilities: File, Edit, View, and Help. This menu bar is in addition to the menu bars provided by tools.

R4b.2 The File menu shall contain capabilities to

- a. open a proposal file (an XML file)
- b. open a template file (an XML file)
- c. save a proposal or template file
- d. exit session

R4b.3 The Edit menu shall contain

- preferences for working directory, e.g., color, size of APT on screen, etc
- Capabilities to edit the hierarchical view (HEM design only).

R4b.4 The View menu shall control the screen layout such as

- Shrink toolbar (eliminates graphical icons)
- Not display the mini-spreadsheet

R4b.5 The Help menu shall contain Help information about the APT in general and provide access to individual tool help.

R4b.6 Help shall contain directed examples that display the sequence of steps required to carry out some common tasks.

c. Undo and Redo

R4c.1 There shall be an Undo button that reverses the actions taken by tools in making changes to a proposal. Undo can be applied several times to backtrack to some earlier point in the proposal development. The Undo operations can reverse the state of the proposal as far back as the start of the current APT session. Restrictions within tools may limit the extent to which undo can reverse actions.

R4c.2 There shall be a Redo button that reverses the action of Undo.

d. Tool Bar

R4d.1 There shall be a tool bar that lists the available tools, both interactive and batch.

R4d.2 For Phase 1, the tool bar shall provide for the
Hierarchical Editor (in the HE Tool design only)
Proposal Editor (includes Proposal Description Editor and Spreadsheet Editors)
Visual Target Tuner (Includes Duplication Checker, Bright Object Tool, and Archival Research)
Exposure Time Calculators
Resource Estimator
Batch Check Tool
Submission Tool

R4d.3 For Phase 2, the toolbar shall provide for the
Hierarchical Editor (HET design only)
Program Editor (includes Program Description Editor and Spreadsheet Editors)
Visual Target Tuner (includes Duplication Checker, Bright Object Tool, and Archival Research)
Exposure Time Calculators
Orbit Planner
Visit Planner
Batch Check Tool
Submission Tool

R4d.4 The tool bar shall use
static GIF images for buttons
a concise text label
tool tips

e. Hierarchical Viewer

R4e.1 There shall be a hierarchical viewer that lists proposal items and templates.

R4e.2 The hierarchy of items for Phase 1 shall consist of
proposals
 proposal description
 observations
templates

R4e.3 The hierarchy of proposal items for Phase 2 shall consist of
proposals
 proposal description

- targets
- visits
 - exposures
- patterns
- templates

The hierarchy implies that exposures must be created within visits. However, users can create exposure templates that do not belong to a visit. Templates belong to a Phase and not to a particular proposal.

R4e.3a Templates shall be listed in the hierarchical editor, but they are not actual proposal items.

R4e.3b Users shall be able to work on items from more than one proposals in a single session. In some cases, a single tool shall be able to operate on items from more than one proposal.

R4e.4 In the hierarchical editor, the naming conventions of items shall follow a simple consecutive numbering scheme, Target1..., Visit1..., Exp1..., Pattern1. Deleted proposal items may leave holes in the list.

R4e.4a In the hierarchical editor, newly created proposal visits, exposures, and patterns shall be named Targ<N>, Visit<N>, Exp<N>, and Pattern<N> respectively, where N is one greater than value for any target in the case of targets, any visit in the case of visits, any exposure with the same visit in the case of exposures, and any pattern in the case of patterns.

R4e.4b The format for targets in the hierarchical editor shall be TargN(Target name). If the target name is unknown, it shall be omitted.

R4e.4c The format for exposures in the hierarchical editor shall be ExpN(Target name, Pattern). If either target or pattern is unknown, it shall be omitted.

R4e.5 There shall be a mini-spreadsheet that is read/write.

R4e.5a The mini-spreadsheet shall be displayed as a scrolled region.

R4e.5b The mini-spreadsheet shall display information about proposal items that are selected in the hierarchical viewer.

R4e.5c Any changes made in the mini-spreadsheet shall be immediately reflected in the currently displayed tool, where appropriate.

R4e.6 A mouse click on a proposal item in the hierarchical viewer shall result in the item being "selected" and the following shall occur.

- a. The proposal item shall be highlighted.
- b. A row shall appear in the mini-spreadsheet, which contains the same details as would appear in the full spreadsheet editor for that item.
- c. The tools in the toolbar that are not appropriate for use by this item shall be made inaccessible. That is, their buttons shall be less visible and they shall not be selectable for use.
- d. The proposal item shall provide a “context” for the tool subsequently selected. That is, the tool shall work upon the selected proposal item.

R4e.7 More than one proposal item can be selected, even if the items span proposals. Individual tools may not be able to operate on a set of selected items simultaneously. It is the responsibility of the individual tools to define the appropriate action.

R4e.8 Each type of proposal item and template in the hierarchical viewer, except proposal description, shall contain a <new> selection.

R4e.9 If the user applies a mouseclick on <new>, then another item of that type shall be created in the hierarchy, which follows the naming conventions described in R4e.4 to R4e.4c.

R4e.10 The detailed information about that item shall be largely blank. In the case of <new> exposures, the visit number is known.

R4e.11 The hierarchical viewer shall be visible at all times.

R4e.12 Object help shall be available on each proposal item. This help can contain a menu of tools that can operate on the item.

R4e.13 The viewer workspace shall include View and Help menu items.

e1. Hierarchical Editor for HE Mode (HEM) Design

Re1.1 The hierarchical editor shall share the same workspace as the hierarchical viewer.

Re1.2 A means shall be provided to clearly distinguish which of the two modes (view or edit) is in operation and to allow the user to select the mode.

Re1.3 The view mode shall automatically be in effect whenever a tool is active.

Re1.4 Menus for the hierarchical editor are provided by GUI main menus.

Re1.5 Changes to the proposal structure may cause the tool display to be out of date. There shall be a means of conveying this inconsistency to the user, such as greying out the tool display.

e2. Hierarchical Editor in HET Design

Re2.1 The proposal viewer work area shall be used for browsing only and not for general editing.

Re2.2 The hierarchical editor shall operate like all other tools and display information in the tool area.

Re2.3 The menus for the hierarchical editor are contained in the tool area.

f. Templates

R4f.1 Users shall be able to save results from tools without being directly part of the proposal. The items, which save this information, are called templates.

R4f.2 Users shall be able to apply templates to supply information to proposal items.

R4f.3 Two general categories of templates shall be provided: proposal item templates and tool templates.

R4f.4 Users shall be able to save proposal item information in proposal item templates for the following items: observations in Phase 1 proposals and targets, visits, exposures, and patterns in Phase 2 proposals. Proposal item templates shall have the same structure as the corresponding proposal items.

R4f.4a Proposal item templates shall follow the naming convention <type>_template<number>, such as exp_template1.

R4f.5 Users shall be able to save information provided by a tool in tool templates. Tool templates are aggregate items consisting of proposal item templates.

R4f.5a Tool templates shall follow the naming convention <tool> template<number>, such as vtt template1.

R4f.6 Users shall be able to apply templates to proposal items by copying the template to compatible proposal item(s) in the hierarchical editor (see R4e.7).

R4f.7 Users shall be able to copy data from proposal item templates into proposal items of the same type, i.e., exposure templates can be saved into exposure items, but not into visit items.

R4f.8 Users shall be able to save data from a tool template into one or more members of a set of proposal items of the type contained within the aggregate. For example, VTT templates contain exposure, target, and visit templates. They can be saved into exposure, target and visit proposal items.

R4f.10 There shall be a tool that takes as input Phase 1 proposals and generates as output Phase 2 proposal item templates. This tool could be used to provide successful Phase 1 proposers with a set of templates to start their Phase 2 proposal development.

R4f.11 Users shall be able to create templates as part of proposal item or tool item (such as a VTT aperture) help. That is, it shall be possible to create templates by clicking on the object with a control-mouseclick to view the possibilities.

R4f.12 Users shall be able to select a template as a context for a tool.

R4f.13 If a value changes within a template, the previously assigned values to proposal items by that template shall not be automatically changed. That is, templates shall only copy values to items, rather than copy pointers to values in templates.

R4f.14 User shall be able to view and edit templates in spreadsheet editors. For proposal item templates, the proposal item spreadsheet editors can be used for this purpose. For tool templates, there could be tool templates spreadsheets. Another possibility would be to allow a tool template to be viewed in the proposal item spreadsheet editors that are appropriate for the type of template being viewed.
See also R4e

g. Tool Management

R4g.1 Tools shall be displayed within a tool area of the APT.

R4g.2 Users shall be able to display only one tool within the tool area.

R4g.5 All tools except the proposal submission tool shall run synchronously. That is, a tool must complete processing before another tool operate or the same tool can operate again.

R4g.6 A particular mouseclick on certain tool objects displayed within a tool shall result in a help menu for the list of possible actions on that object. For example,

clicking on a VTT observation shall result in a list of possible actions involving that observation, which include saving the aperture as a template of various forms, applying the observation to a visit, and changing the observation orientation.

R4g.7 Users shall be able to enter a tool with or without having selected a context.

R4g.7a User shall be able to select no context by some appropriate convention, such as clicking on currently selected proposal items and templates (i.e., select works as a toggle).

R4g.8 Users shall be able to enter a tool without a context. Objects created within tools shall be automatically assigned to new proposal items. The details are TBD.

R4g.8a If any rules are violated (such as changing an orient on a single VTT exposure), the user shall be warned of the implications.

R4g.9 Users shall be able to run interactive tools in any order.

R4g.9a Users must successfully run the Batch Check Tool before Submitting a Phase 2 proposal. The APT shall verify that this check is done before the proposal is submitted.

R4g.10 By default, a context shall persist as different tools are selected.

R4g.11 Users shall be able to change context at any time. If context is changed while a graphical tool is being displayed, the tool shall immediately switch to the new context.

h. Tool Conventions

R4h.1 Each interactive tool shall have its own main menu toolbar.

R4h.1a Through the Edit menu within some tools, users shall be able to save results files and open results files that apply to that tool.

R4h.2 The VTT, ETC, and Starview2 tools shall be able to run standalone. The tool in standalone mode shall be nearly identical in APT mode.

R4h.3 Each tool shall provide object help as defined in 4e.7.

R4h.4 Each tool shall provide help that is also accessible by the APT help.

R4h.5 Each tool shall perform checks of data as soon as possible after entry. The APT shall perform checks of proposal data, including checks that span tools, through the Batch Check Tool.

i. Proposal and Template Files

R4i.1 Information about proposals and templates shall be saved in files.

R4i.2 The files are read and written by the APT by means of the File main menu.

R4i.3 The format for these files shall be XML.

R4i.4 There shall be a tool that converts RPS2 files to XML proposal files.

R4i.5 The files shall contain all information required to specify a complete proposal. In addition, it contains auxiliary information that is used by the APT, such as the type of spectrum used for an exposure by the ETC.

j. Interactions with external environment

R4j.1 A user shall be able to copy text from any text i/o area internal or external to the APT to any text area within the APT, using simple mouse conventions that are consistent with the OS. For example, it should be possible to copy text from within a Solaris window, following the usual conventions of the left mouse button double-click or drag, to any text area within the APT GUI, including any APT tool, following the convention of middle mouse button click. Special characters such as carriage returns and new lines will be handled properly.

R4j.2 A user shall be able to copy APT text from any text i/o area to any text area outside APT, using simple mouse conventions that are consistent with the OS.

