APT: HST Proposal Preparation Environment for the Second Decade

A. Koratkar, N. Bernstein, R. Douglas, A. Gerb, J. Jones, K. Peterson, R. van der Marel

August 19, 1999

EXECUTIVE SUMMARY

The Tools of the Future Working Group was formed to study improvements to the proposal preparation tools we offer our user community (scientists and operation specialists), with two goals: (1) to facilitate scientific investigation for observers, and (2) to decrease the effort spent on routine matters by observatory staff. In this report, we propose:

1. A modern set of proposal tools and an environment that integrates them. Compared to existing RPS2 software, the proposed software will be more intuitive, visual, responsive, interoperable, and extensible to other observatories.

2. A management structure that supports innovation. We propose to divide development activities into *innovating* and *fielding* efforts to prevent operational pressures from inhibiting innovation. This will allow us to use of up-to-date technology, and to remain fluid and responsive to changes. This strategy will ensure state-of-art tools for proposal preparation for the user community.

3. A delivery plan that puts the earliest tools in the hands of the users in time for the next Phase I and II proposal submission cycles. We plan continued frequent deliveries of new tools and capabilities over the next twenty-four months.

APT, the Astronomer's Proposal Tools, is our vision of the proposal preparation tools environment.

1. Charter

STScI's overriding goal of maximizing the scientific returns in a resource limited environment requires us to consider innovative approaches to user support. The Tools of the Future Working Group was chartered to envision an integrated tools environment that will achieve the following user support goals:

- Use of state of the art technology to allow observers to primarily spend time on scientific decisions and not on the mechanics of using the system.

- Have observers produce observations that do not require any manual intervention by our staff at least 80% of the time.
- Ensure time for ongoing innovation by freeing staff from repetitive tasks.

In particular, the group was to concentrate on a tools environment for the proposal preparation process that makes it easy to turn an idea into an observation. In this report we present:

- the evolution of proposal preparation tools at STScI (section 2),
- the APT tools and integrated environment (section 3),
- the implementation timeline (section 4),
- the architecture (section 5),
- the management strategy (section 6), and
- the costs and resources (section 7)

2. Evolution of Proposal Preparation Tools at STScI

STScI has invested heavily in tools that can be used during proposal preparation. These tools have evolved over time in an attempt to keep up with both technology and observatory operations strategy.

2.1 History

Originally, Hubble proposals were submitted using the Remote Proposal Submission System (RPSS). This submission system was designed in the mid-1980s and represented the state of technology and experience in handling “service mode” observations. It provided the bare minimum of user support, for example, checking for syntax or spelling errors and some illegal configurations. When received at STScI, these proposals were processed to determine feasibility and schedulability. It was at this stage that most problems were discovered, and manual intervention by operations staff was necessary. RPSS was used until 1994. RPSS was neither conducive to efficient user support strategies nor was it observer/observatory staff user friendly.

After three years of proposal preparation experience, an effort was initiated to improve upon the RPSS process, which led to the release of RPS2 for cycle 5 (1994) observing. RPS2 was designed to further two major goals:

- Improve the quality of proposals at submission time and thereby avoid the need for observers and operations staff to iterate. This was achieved by making some of the telescope operations constraints available to observers when they prepared their programs.
- Make routine the process of updating proposals after submission for scientific or operational reasons. This was achieved by dividing proposals into “visits” which can be independently planned and scheduled.

RPS2 was implemented using client/server technology which processed a proposal in batch, and then graphically displayed the results of the processing to the observer. In developing RPS2, we opted in favor of the modest RPS2 architecture instead of a full interactive environment because of the shortcomings of Trans (there was no way to query Trans to receive a quick answer to a question) and the lack of suitable integration software like Corba and Java RMI which were still years away. Further, the simpler RPS2 architecture, which was a vast improvement over RPSS, could be provided to the user community quite rapidly. A fully interactive system was thus not cost effective in 1994.

Evaluated against its original implementation goals, RPS2 has been a success. The vast majority of proposals are submitted today without feasibility or schedulability errors because observers are given sufficient information to remove these errors during proposal development. Updating observation parameters shortly before execution is now simple and commonplace. However, there are major areas with potential for improvement, as we detail in subsequent sections.

2.2 An Example Contrasting RPS2 and a Modern Tools Environment

The following simple example illustrates the advantages of the proposed new environment over RPS2.

Example: A proposer wants to mosaic a large area of the sky (e.g. a cluster of galaxies) with WFPC2.

We compare below the steps taken by the proposer to achieve this activity. In our present RPS2 proposal preparation strategy there are a number of inefficiencies, both for the proposer and observatory staff, which can be either eliminated or reduced with modern tools and a unifying environment.

	Problem solved in RPS2	Problem solved in the new environment
1	Proposer writes his/her own code to determine how the WFPC2 L-shape can be used to tile a region of the sky.	Proposer uses the Visual Target Tuner (see section 3.3) to display an image of the area of the sky of interest. This may be a DSS image, or a ground-based image provided by the observer. Proposer clicks on several points that mark the region that needs to be mosaiced.

	Problem solved in RPS2	Problem solved in the new environment
2	Proposer guesses a possible orientation for the observations. Based on this orientation proposer specifies n positions (targets or POS TARGs), one for each tiling position. These are then entered by hand on the phase II proposal.	Software tells proposer whether there is an orientation that yields guide stars and whether there are bright objects to avoid. It then recommends the most efficient orientation, and displays the tiling on the sky. Proposer accepts the software recommendation. If the recommendation is unacceptable, the proposer interactively tries personal mosaicing strategies and determines the schedulability of the mosaic. Proposer approves a mosaic pattern, and the target/aperture positions, available guide stars and other relevant information that lead to the scientific decision are recorded in the Phase II.
3	The proposer completes the rest of the Phase II. For each position of the mosaic separate visits/exposures are defined by hand, despite the fact that the same observation sequence is requested for each target position.	The proposer completes the rest of the Phase II. The required observation sequence has to be listed once, and then it is replicated for the entire mosaic pattern.
4	Proposer submits Phase II.	Proposer submits Phase II.
5	PC studies Phase II and possibly discovers there are no guide stars for the given orientation and tells this to the observer.	No further proposer-Program Coordinator (PC)-Contact Scientist (CS) interaction is required, and observations are executed.
6	Proposer *rewrites the whole Phase II* for a new assumed orientation and *re-submits.*	
7	PC studies Phase II *for the second time* and finds that there are guide stars. PC forwards proposal to the CS, who may determines that some bright objects will create calibration/photometric problems, and tells this to the observer.	
8	Proposers *rewrites the whole Phase II* for the third time for a new assumed orientation which eliminates/reduces the problems raised by the CS, and then once again *re-submits the proposal.*	

	Problem solved in RPS2	Problem solved in the new environment
9	PC and CS study proposal *once again*, and determine that the proposal is now schedulable and will not cause any major problems. PC generates finder charts. The finder charts *do not show* FOV overlays, so the *proposer cannot properly determine whether the positions and coordinates really cover the area to be mosaiced.*	
10	The proposer makes a judgement call, accepts the galley proof and observations are obtained. Despite all iterations, there have been cases where the final data, incorrectly covers or did not cover the intended sky area.	

3. APT

In the last three years, technological advances such as widespread use of the Internet, multi-platform visual development tools, and overall increases in the power of desktop hardware are allowing for significant improvements in user support tools that can be provided by an observatory. **APT, the Astronomer's Proposal Tools,** is our vision of the integrated environment that will:

- Leverage off state of the art technologies;
- Provide modern user support tools; and
- Achieve the goals stated in section 1.

APT consists of two major components: the *APT tools set* that provides users with tools that are more intuitive, visual, and responsive, and the *APT integrated environment* that unifies all the tools and makes them interoperable. APT will provide capabilities that cannot be made available in existing RPS2-based software.

In designing APT and planning the implementation/management strategy we have applied lessons learned from the NGST Scientist's Expert Assistant (SEA) prototyping effort. The objective of the SEA project was to develop and evaluate visual and expert system tools to determine if they can dramatically reduce the amount of manual effort that currently goes into the present proposal process. The HST Phase II process was selected as a testbed to evaluate the effectiveness of SEA. For more details on SEA see http://aaaprod.gsfc.nasa.gov/SEA/. Although the SEA project has not completed its evaluation, we believe that the project has successfully prototyped a number of tools and concepts. Thus, applying lessons learned from SEA is appropriate.

3.1 - High Level Goals for APT

For developing the APT environment we considered the following high level goals. The tools and environment will:

- Make the proposal preparation process more intuitive and observatory operations less cumbersome;
- Provide documentation/help that is friendly, up to date, and easily accessible to users of varying levels of expertise; and
- Define an extensible framework which is responsive to changes in both technology and observatory operations.

3.2 Some Basic Development Guidelines for the APT Integrated Environment

In developing the vision for APT we have used many of the guidelines suggested in the white paper “A New Paradigm for User Support and Software Tools” by G. Miller, A. Koratkar, & D. Golombek, ITM-1999-03. The APT integrated environment would be designed to provide the following:

1. ***User orientation*** - All components of the system will use terms and concepts which are meaningful to astronomers. This will allow the tools to be friendly and intuitive at first glance.
2. ***Responsiveness and speed*** - Whenever possible, results of user actions will be available immediately. Instantaneous graphical updates when changes are made will be presented to users whenever possible. This responsiveness will reduce user frustration associated with batch processing. It will also allow the user to make scientific trade-offs in a timely manner.
3. ***Scientific feedback*** - The system will provide information needed to make scientific trade-offs. The impact of a choice will be shown in a meaningful way, and thus users will be made self-sufficient.
4. ***Easy use -*** To accommodate user expertise and preferences, proposal information can be accepted in multiple formats. For example, multiple views will be available for all proposal information, including graphical, table form and ASCII representation. “Wizards”- dialogues which guide users through common tasks will be available if requested by the user.
5. ***Easy installation -*** Straight forward web-based installation with a highly-portable, platform-independent implementation.
6. ***Uniformity*** - All tools will use consistent terminologies and have a similar “look and feel” to reduce the learning curve. We will take care to adopt the same user

interface standards that are widely used in general purpose software and other tools used by the astronomical community.

7. ***Interoperability*** - Tools will be able to share information, alleviating users from having to manually enter and re-enter data and re-process information. Manipulating proposal data should be easy.

8. ***Useful documentation*** - Documentation will be an integral part of the toolset and will be structured to allow efficient access by humans *and* software tools. By providing documentation/help that is friendly, up-to-date, and easily accessible to users of varying levels of expertise. Thus, we will reduce the effort required by observers and observatory staff to solve routine problems.

9. ***Common environment*** - Observers will have access to the same tools environment and configuration as observatory staff.

10. ***Extensible framework*** -An object-oriented mission-independent architecture will be designed to be easily specialized for other observatories. An object oriented architecture has the advantage that the basic class libraries can be used by other applications not associated with proposal preparation.

11. ***Open architecture*** - It will be easy for everyone to enhance tools or include fixes ("open source" software paradigm). The initial system created and developed will work like a kernel around which the facility will grow following the needs of the community.

12. ***Capturing expert knowledge*** - We will explore methods which capture human expertise and make it usable to humans and software.

In the APT integrated environment we propose a software library of reusable components for use in astronomy. The flexible, extensible framework and class libraries can be used by other applications to display and manipulate images, catalogs or other astronomical data in various formats. The source code will be freely available and developers will be encouraged to contribute new components and suggestions. This will allow us to leverage off the efforts of other groups in the community. Ideally, in such a common tools framework it should be easy for observatories to share observing software components, and to extend existing components to meet unique needs. Observatory software support is rapidly evolving in this direction. As a consequence of the Workshop on Observing tools (see http://aaaprod.gsfc.nasa.gov/workshop/WorkshopReport.htm), ESO is already discussing a collaboration with us for creating such a library (see http://archive.eso.org/JSky/) of tools for visualizing an observatory's field-of-view. Also, within STScI we have other divisions working on Java tools that can be integrated with APT; for example DSD is developing tools to access the archives. These can be integrated with APT so that users can access archives effectively during the proposal preparation phase. To achieve success in leveraging other groups' efforts, there will have to be more dialog between the various groups involved. The management strategy developed below suggests a possible way for opening dialog and thus paving the way for sharing of code.

3.3 Tools we would like in APT

In determining the tools and functionalities that we would like to have in the APT environment we not only considered our high level goals, but also evaluated the tools for the impact that they would have on both the scientific community as well as observatory staff. By and large, the tools had to provide large improvement over RPS2 tools. Some tools have been suggested to complete the integrated environment so that the whole proposal submission process is possible in a single environment. The following lists the tools and functionalities that we consider as the basic set we would like to have in APT. The tools can be extended to provide more detailed functionality as both the tool and the environment mature.

1. ***Visual Target Tuner (VTT)*** - We would like to prepare the prototype SEA VTT for operational release and use. Currently there is no tool that allows proposers to visually determine the exact field of view that is appropriate for their science. Availability of such a tool would not only provide proposers with information earlier in the proposal preparation process, but it will also reduce observatory staff effort that is presently being spent on iterating over details of a proposal with the proposer (see the example in section 2.2). At present, the SEA VTT does not provide useful information concerning available guide stars. We consider this a promising functionality to be developed.

   Other candidates for improved functionality are access to data sets in archives, display of offset patterns, bad pixel information, connection to IRAF/STSDAS, improved access to target catalogues and lists, and ability to represent spectral lines, grisms and coronography.

2. ***Exposure Time Calculators (ETCs)*** - Web-based ETCs already exist and are extensively used by observers. The prototype SEA ETC tool is the next step towards ETCs that provides users with the capability to effectively explore the available parameter space. We would like to have such an interactive ETC in the APT environment. The ETC is an important tool to integrate into the environment as it can provide easy access to a functionality that is always being used by proposers as they develop observations. A logical choice would be the SEA ETC for operational release and use.

3. ***Phase I Submission Form*** - We would like to provide Phase I proposers with a web based electronic form to simplify the submission process. This will likely be implemented using the SEA proposal definition forms.

4. ***Exposure Planner*** - Presently all users expend a lot of resources in laying out their exposures in the allocated orbits. This task in RPS2 is time consuming and frustrating. We would like to continue to develop the prototype Exposure Planner developed by the SEA group that displays exposures as they will be executed within orbits. It allows manipulation of exposure times and ordering with instantaneous updates of overhead information. This will make it easier for observers to lay out their orbits without time-consuming iteration with RPS2. The earliest

versions will be based on a rough estimate of overhead times matching those described in the Phase I Call For Proposals.

As the TransVERSE project matures, we would like to implement the capability to connect to TransVERSE and receive far more accurate information including inserted parallel observations and buffer management. We would also like to explore allowing users access to a detailed breakdown of overhead components. Once later phases of TransVERSE are complete we would like to implement an optimizer under the control of the observer which uses TransVERSE's search capabilities to improve the efficiency of the observing program.

5. ***Bright Object Checker*** - Bright object checking is an essential part of our Phase II process which directly affects the health and safety of our instruments. At present, observatory staff do all the bright object checking (often manually) and once again spend time on iterating over details with the proposer. We would like our software to provide information about bright objects to observers. This will help decrease the amount of work done at STScI after submission to address bright object issues. Since the New Guide Star System (NGSS) is the most accurate source of bright object information, this capability is likely to be implemented via a connection to NGSS.

6. ***Visit Planner*** - We would like to develop a graphical tool that allows observers to visualize timing relationships between visits (e.g. BEFORE, AFTER, GROUP WITHIN) and to better understand unschedulable situations. At present there is no visualization of timing links and other schedulability information. between observations to determine the effect of a change on the rest of the program. A connection to the Spike system will allow such a tool to provide instantaneous schedulability feedback.

7. ***Canned Observing Strategies*** - We would like to automate the process of applying customizable observing strategies to observing programs (e.g. mosaicing). In RPS2 such a task is cumbersome.

8. ***Import of Data from Archives*** - We would like to be able to import details of an observing program from any of a number of mission archives to be used to form a new proposal. The goal here is to support the proposal development process by allowing observers to graphically visualize data. We would most likely start with the Hubble archive and add access to other archives as feedback and feasibility indicate.

9. ***Improved Software Updates*** - We would like to improve the way observers have access to the latest data on the state of the observatory. We need a strategy that will allow up-to-the-minute access to operational and hardware changes, but that also supports those who wish their environment to remain stable while they compare the results of scientific trade-offs.

10. ***Tight Integration with Online Documentation*** - We would like to couple our automated tools with online documentation so that information on any part of the system is easy to find.

11. ***Access to Execution Data*** - We would like our observer tools to be able to access operational data. This would be useful, for example, in making schedulability determinations based on exactly when observations have executed or will execute. Such a capability would reduce effort to implement proposals at STScI by decreasing the incidence of unschedulable observations due to execution information now unavailable to RPS2.

12. ***Grouping Observations for Global Update*** - We would like our tool to allow a proposer to group observations to perform a single update to all of them, such as a filter change or new target. If, for example, an observer finds out at a late stage that a planned target is infeasible, it should be easy to substitute another target without a great deal of search-and-change effort.

4. Implementation Timeline

Based on the overwhelmingly positive response to the tools prototyped in the SEA, we plan to offer new functionality as early as possible in the APT release process. *This strategy dictates releasing tools that do not yet handle the myriad types of observations that can be proposed for HST.*

During this initial stage, RPS2 and the APT Tools will run independently, but cooperatively. Each tool will be capable of producing output that could be incorporated into an existing RPS2 file, and then be processed with RPS2.

With the creation of the APT Integrated Environment, the output of the newly developed tools will be able to be collated into a complete proposal within APT itself. One of the output products of APT will be a complete RPS2 file.

Next, APT will begin to assume capabilities (syntax, feasibility and schedulability checking) that make existing RPS2 tasks obsolete. At this stage many users may find that RPS2 is no longer required to submit a proposal to STScI. However, it is expected that some observers will still find it necessary to use RPS2 to produce their proposals. Until all RPS2 functionality is incorporated into APT users will be able to use either system, though it is expected that they will prefer to use APT when possible.

In the previous section we have described many tools that we would like to see in APT. Of these many tools, some have been prototyped in SEA. We have a reasonable idea of the impact of these tools. For the other tools, it is far from clear which tools will make the most impact. Therefore, we will introduce these capabilities using a prototyping methodology, wherein we release mock-ups and receive feedback from our user community as to their utility. As it is impossible to know which tools will be deemed most desirable by the users, the priority of releasing each tool or environment enhancement is likely to change. Because of this variability it is premature to predict when RPS2's functionality will be completely subsumed by the SEA. We may eventually want to drop some tools in favor of others. *Our intention is to remain fluid and responsive to ideas and not to adhere blindly to this plan as more information becomes available.*

In the ensuing sections we divide the tools and functionalities that we described in the previous section into four stages for deployment. Note that *prototyping will begin immediately and will continue throughout the process.*

- ***Stage I - Delivery in Fall-Winter 1999:*** This consists of operational releases of some tools prototyped in SEA, and prototypes of new functionalities. This release of tools and functionalities will supplement RPS2 in time for cycle 9 Phase II or cycle 10 Phase I proposal development. Work on these has already begun.
- ***Stage II - Delivery in Spring-Fall 2000:*** This consists of those functionalities we expect to release operationally or prototype in the next year. In this stage we would have a prototype release of the integrated environment for Cycle 10 Phase II. Work on these should begin immediately.
- ***Stage III - Delivery Beyond Fall 2000:*** Includes operational release of the integrated environment. Work on this will have begun by winter 1999.
- ***Promising Avenues for Study - Delivery Beyond Spring 2001:*** Further enhancements to APT and re-evaluation of other promising ideas.

4.1 Stage I: Delivery in Fall-Winter 1999

We have already begun work on promising tools that were prototyped in the SEA effort. This will allow us to provide the user community with effective modern tools as early as cycle 9 Phase II development and cycle 10 Phase I development. These tools are operational versions of the SEA prototype tools. The emphasis is on sending out the tools as quickly as possible to the user community. Therefore, these first versions of the tools will have little in the way of additional functionality compared to what is currently available in the SEA prototype. The SEA tools that we have slated for operational release are:

1. *the Visual Target Tuner*
2. *the Exposure Time Calculator*
3. *the Phase I submission tool using SEA forms.*

Following our philosophy of continuous innovation, we will simultaneously be developing prototypes. The prototypes slated for evaluation in this stage are:

1. A server for external *availability of guide star information*. RPS2 cycle 9 will be used as a test-bed for the NGSS server.
2. A *Phase I Resource Estimator,* which will use the computation algorithms in the Call for Proposals. Intended as an alternative to cycle 10 Phase I hand calculations.
3. An *Exposure Planner*, which graphically displays the information from the resource estimator.

4.2 Stage II - Delivery in Spring-Fall 2000

Operational releases in this stage are:

1. A *Visual Target Tuner* update, with new capabilities chosen from those listed in the VTT item above (e.g. display of selected guide stars, etc.).
2. Improvements to the *Exposure Planner* based on reactions to the prototype.

Prototype developments in this stage are:

1. *The Integrated Environment*.
2. A *Bright Object Checker* which will provide information to the observer, possibly via the VTT.
3. An *Exposure Planner* that is connect to the Trans system using the new capabilities of TransVERSE.
4. A *Visit Planner* with a connection to the Spike software to provide instantaneous schedulability feedback.
5. Connection to the status server at STScI.
6. “Canned” exposure strategies.

4.3 Stage III - Delivery beyond Fall 2000

Operational releases slated for this stage are:

1. Improvements to the *APT integrated environment,* designed to limit reliance on RPS2 and to address issues raised during use of the prototype.
2. An improved *strategy for handling software and observatory parameter update*s.
3. Improvements to the *Bright Object Checker* based on reactions to the prototype.
4. Improvements to the *Visit Planner* based on reactions to the prototype.

Prototype developments in this stage are:

1. An optimizer for the Exposure Planner using communications with TransVERSE.
2. Improved access to online documentation to take advantage of a reorganized website.
3. Ability to import details of a proposal from mission archives.
4. Ability to group observations for batch update.
5. An exploration of what it will take to completely eliminate RPS2 from operational use.

4.4 Promising Avenues for Study - Delivery Beyond Spring 2001

We will be considering other promising avenues for study, and at intervals we will evaluate these to determine their usability in APT. We think that the technology for making these functionalities effective is still in its infancy. The promising avenues are:

1. Making the source code available via an open-source license allowing other centers to build on our work.
2. The ability to “tune” wizards so the information they give can be varied depending on the needs of the user.
3. The ability for the system to record not only “what” the observer wants, but also “why” the observer wants it. This can help direct operational recovery efforts for observing programs later found to be infeasible.
4. Inclusion of data “pedigree” records that list where data came from. (Catalog? Selected from VTT display? Entered by hand?)
5. Data mining to discover scientific knowledge in existing databases for application to a new science program.
6. Giving the software the capability over time to “learn” how best to support a given user.

5. Architecture

In order to remain up-to-date with technology, and to have a system which is extensible and generic, we have devised an architecture which is vastly different from that used by RPS2. The core differences are in the sharing of data between tools and the emphasis on interactive rather than batch processing.

5.1 Design Principles

In order to ensure that the system is extensible and easy to maintain and develop, the system will be developed under the following principles:

- We will favor interactive approaches over batch processes. Performance and responsiveness will be considered in all design decisions.
- In order to provide a useful system as early as possible, we will reuse existing components where appropriate (e.g. TransVERSE).
- The design will be completely object-oriented.
- In order to grow towards an observatory-independent system, the design will be documented so that it is easily understood (e.g. using UML).
- The system will be extensible, i.e., it will be easy to add new tools.
- We will use a platform independent language wherever possible. Java is the leading candidate because it provides a number of advantages such as:
    - interactive graphics and image processing support;
    - support for accessing databases and catalog servers;
    - user interface support;
    - portability; and
    - object oriented development.

- The system will use a common data format. The markup language XML is the leading candidate since it is a world wide standard for data representation. The advantages of XML are:
    - support of automatic checking of documents for structure validity;
    - availability of a rich array of tools to process and display XML documents; and
    - availability of existing Java libraries that already support XML.

5.2 High-level Architecture

The goal of this system is to provide an environment for performing the tasks necessary to develop a scientifically useful and feasible observing proposal. One of the goals of the architecture is to support a wide range of tools. Not all the tools are available now or will necessarily be created or maintained by the Institute. The Environment is designed to maximize communication and data sharing among a variety of tools. Observers will be able to manipulate their observations in ways that makes sense to them, and that depend on the observatory they are using.

The architecture is divided into five parts: the Graphical Environment, the Communications Infrastructure, the Data Pool, the Tools, and the Wizards.

1. ***The Graphical Environment*** provides access to the various tools in a seamless way. Each tool will have its own graphical user interface, but the environment unifies them. The Graphical Environment will provide a standard set of features that the tools may utilize, such as Clipboard support (Copy/Paste).

2. ***The Communications Infrastructure*** is the means by which tools communicate with each other and the Graphical Environment. It can even connect tools which are written in different languages.

3. ***The Data Pool*** consists of those objects which the various tools will share. It contains all shared objects that comprise the Proposal, Targets, Exposures, and Groups. Some objects are kept internal to an individual tool, while others are shared. The shared objects are well-defined. The Communications Infrastructure makes it possible to share these objects across various tools.

4. ***The Tools*** are all individually written and maintained, with their own architecture and data concerns. They manipulate objects in the data pool, as well as their own unshared objects. The interfaces to the data pool are controlled by the data pool part of the architecture. The tools provide ways of visualizing the data pool and manipulating it. Each tool will provide a help mechanism to answer questions about the use of that particular tool. Some tools may be extremely simple, while others may be highly complex.

5. ***The Wizards*** are separate pieces of software which either can be invoked or can react to actions by the user. Wizards are programmed with knowledge about the tools and the tasks that one might wish to perform. They can ask questions and run tools in order to help the user follow recommended strategies for observing. In reactive mode, Wizards will offer to help when they recognize a particular

strategy that is being used or when they see a suboptimal choice being made. Wizards can be configured to give more or less information, based on the user's preference and expertise.

6. Management Strategy

We would like to organize development of APT around the two goals of innovation and frequent improvement.

6.1 Innovation

We would like to continue applying the latest advances in software tools to make tangible improvements in the lives of our observers. It is not acceptable for us to release a tool and perennially polish it without adding substantial new capabilities.

6.1.1 Obstacles to Innovation

Our management plan is geared to overcome the following obstacles to innovation:

- *Demands of Operational Support* - Working closely with operational users has been a great benefit in many ways, but it also has tended to make it hard to free up sufficient resources to develop new user tools. We find our resources completely taken up with developing worthwhile improvements to operational software and find it hard to reserve time to develop new prototypes.
- *No Room for Failure* - Coming up with good ideas necessarily implies coming up with bad ideas as well. It's not always obvious from the start which ideas are good and which aren't. Before the VTT was developed, the SEA group spent a lot of time researching an "Interview Mode" which developed a scientific program based on the answers to a dialogue. This proved to be a dead end (though some of what was learned was later used to define the ETCs), but did not preclude later successes by the group. In the past, setting aside resources for exploration in the Presto Software Support Team has typically been a high visibility affair. Operations groups are eager to get developers back working on their worthwhile projects, leaving little room for false starts, dead ends and learning from mistakes.
- *Ebb and Flow of Major Projects* - The principal reason why RPS2's architecture did not allow it to grow to reach its potential was that not enough growth planning was done. It was originally thought of as a prototype; as something that could be fielded quickly to provide immediate relief to observers. Once the decision was made in Feb. 1994 to deliver it in time for cycle 5 (necessitating a Sept. 1994 delivery), all resources had to be aimed at meeting that deadline. We no longer could afford to plan out a growth path at the same time. It was a worthwhile goal to put an improved system in the hands of the observers as soon as possible. But to avoid stagnation we also need a way to keep an eye further into the future.

6.1.2 Management of Innovation

The development team, consisting of scientists and developers will be divided into two groups, the "Innovators" and the "Fielders."

The people in the innovation team - Innovators

The Innovators will play a role similar to that played by the Goddard team that developed the SEA software. They will be prototyping new tools, integrating early feedback, and looking at newer technology that might provide a benefit. The Innovator team should not be considered a short term effort to be dissolved at the end of the project. *If STScI is to stay competitive, it needs to continue to push for improvement.*

The successes that we have had in recent years, Spike, RPS2 and SEA were all the result of *a single individual with a vision on how to solve a problem.* It is true that these were realized by a great deal of effort by a number of talented individuals. But *without such a vision, a gathering of the smartest and most creative people may end up just spinning their wheels.* Therefore, the group of innovators should consist of a number of software developers and a “Project Scientist.”

The project scientist should be an astronomer, and not a software developer. We hope that a good deal of the innovation will come from the developers themselves. The project scientist’s job would be to understand the fundamental direction in which our tools need to improve and to make sure the projects taken on by the Innovators move in that direction. The project scientist would also need to co-ordinate scientific user community input with the development of prototypes. Selecting a project scientist will be the most critical decision made in the staffing of this project. The creative process is delicate and poorly understood, even by experts. Some individuals who play this role well in some areas will play it poorly in others. A great deal of thought must go into this decision to make sure the individual is committed to the goals of the project, creative enough to suggest directions of innovation and receptive enough to feedback to refine ideas. Thought also needs to be given to cultivating a successor to make sure the culture of innovation is ongoing.

To determine the performance and success of the prototype tools there will need to be a tester on the innovation team. This person would work with the project scientist and focus on usability of tools rather than correctness and robustness. Much of the work would be on coordinating alpha testing and synthesizing the feedback for the developers.

The people in the fielding team - Fielders

The Fielders will be developers and testers with the sole responsibility for preparing tools for operational release and making changes that support their operational use. They will work with users to address concerns and questions.

The tester on the fielding team would learn the prototype software, and develop tests for correctness, accuracy and robustness. Just before release there would be a beta testing period and this tester would coordinate and synthesize the beta testing feedback. This tester’s focus though will be more on correctness and robustness than usability.

6.2 Frequent Improvements

We would like to provide frequent updates to our observing tools which represent a clear improvement in the eyes of our users. It is not acceptable for our observers to wait a year (or more) between improvements. Ideally we would like to provide something new and useful several times a year.

6.3 Important Guidelines for Success

6.3.1 Management of the Innovation Team

The most critical element of the APT management plan is that the Innovators remain insulated from the demands of operations, i.e.,

- their funding remains fairly constant and is not looked upon as a resource to tap when operational demands get heavy;
- they have the freedom to experiment and to produce something which differs from what they set out to do; and
- their deadlines are driven by what they would like to present for feedback.

6.3.2 Team Size and Group Dynamics

The Innovators should not be allowed to become too lean. Two developers and a project scientist is a bare minimum, with three to five developers a safer number. However, there may be occasions where a single Fielder will do.

- It is expected that developers will periodically move between the two groups. We want to avoid the Innovators becoming an elite group and for them to be too far removed from the realities of operational use.
- It would be inefficient for a team member to split time between being and Innovator and a Fielder.
- The two groups will be working on the same software. It will be beneficial for them to attend each other’s design and code reviews.
- Interoperability and commonality between our tools and those developed elsewhere in STScI or the community requires management support. Groups must be committed to taking the time required to agree on standards.

6.3.3 Quality assurance goals for the tools/functionalities

The quality assurance goals for the released tools/functionalities will depend on whether the release is a prototype release or an operational release.

- If the release is a prototype, the delivery must not only be usable but also be intuitive, and responsive.
- If the release is operational, the delivery must be robust, i.e., give correct answers, clear diagnostics and not crash or get into unusable states.

6.3.4 Success Criteria for the Teams

- The Innovators should be judged more on the positive reaction to their ideas than on whether they deliver exactly what they promise. They must periodically be permitted to fail as long as they also are periodically brilliant.
- The fielders, on the other hand, should be evaluated based on how quickly and reliably their products reach the users and on how responsive they are to problems.

6.3.5 Outsourcing/Outside Contractors

- The Fielders may increase or decrease in size based on support needs. We may want to augment their ranks with consulting resources at times of heavy support.
- We should carefully consider the trade-offs involved in outsourcing the functions of the Innovators.
- To allow us the widest choice of capable contract personnel, we will need flexibility to consider non-standard work arrangements. For example, it will likely be appropriate for contractors to work off-site or even in some cases, offshore.
- If there are occasions when we will want to augment STScI software development staff with outside consultants, it will be important to train STScI personnel on how to manage small scale software contracts.

6.4 STScI Reorganization

There is an STScI reorganization planned during the time period covered by this project. It is critical that the new organization support our management strategy. It is also critical that the transition offer a minimum of disruption to the activities of the team.

7. Resource Cost

7.1 Scientists and Project Scientist

Innovators cannot develop a tools environment without understanding the needs of the end user, i.e. the proposers/scientists/observatory staff. Therefore, the core group of innovators not only consists of software developers, but also scientists. Also, connected to this core group of innovators, we will on a regular basis solicit help from the user community to determine the types of tools that should be plugged into APT. The two most effective ways to obtain this input would be via the STUC sub-committee on user tools, and via regular input from our staff. To ensure regular and timely input from all users we will need 2 FTEs distributed among 5 - 10 people from the user community for the innovation process. This time is not only to support and advise the project scientist, but also to alpha test the recommended tools.

7.2 Developers

Our development estimates were derived from laying down on a timeline the items we plan to deliver. The resulting timeline is attached in Figure 1.

Based on these results we will be able to develop our group's most important ideas with a team of seven software developers. Five of these will be members of the "Innovators" group we discussed in the Management Strategy session. *This is basically the same size of the development group at Goddard that developed the SEA prototypes.* Two other developers will be members of the "Fielders," though we may want to give ourselves flexibility to increase or decrease this team based on operational demand via variable use of consulting resources.

In addition, some of the projects that involve interfacing software maintained by other development groups (e.g. NGSS) may require some design and development effort from those groups to insure compatible integration.

7.3 Testing Coordinators

In order to achieve the performance goals discussed above, we will need two testers. One would work with the innovators and one would work with the fielders. There will however be overlaps and they will work together on projects especially during transition periods.

7.4 Total Resources and Project Management

For the success of this project cross-divisional support is essential. Members of the Science Support Division have already provided scientific input to this project and the division has promised to support this project by providing resources. The following table shows the resources required per year for the duration of the project.

Team	Resource	FTE	Qualifications	Preliminary Source
Project Management	Project Coordinator	1.0	Software Project Management	PRESTO
	Project Scientist	1.0	Astronomer	PRESTO
Innovators	Scientists	1.4	Astronomers to provide scientific and instrument related insight	SSD (~0.35 FTE per major instrument)
	Scientists	[1.0]	Astronomers to advice on usability of prototypes etc.	5-10 Members of the User Community, for example STUC members
	Operations staff	1.0	PC, schedulars, planners, ...	PRESTO, SESD, ...
	Developers	5.0	Software developers	SSD, PRESTO, DSD, ...
	Testing Coordinator	1.0	Data Analyst/PC/Tester	SSD
Fielders	Scientists	0.6	Astronomers to confirm accuracy and reliability	SSD (~0.15 FTE per major instrument)
	Developers	3.0	Software developers	SSD, PRESTO, DSD, ...
	Testing Coordinator	1.0	Data Analyst/PC/Tester	PRESTO
	Total -STScI	**15.0**		

8. Summary

STScI has always led the astronomical community in defining the concept of user support. In the last three years, technological advances such as widespread use of the Internet, multi-platform visual development tools, and overall increases in the power of desktop hardware are allowing for significant improvements in the user support tools that can be provided by an observatory. Hence, we are convinced that the time is ripe for STScI to move to a new paradigm for user support.

Software Development Resource Timeline