



Instrument PM Project: Software Requirements

Author	Date	Comment
Grant Hill	2/10/2008	Original
Grant Hill	9/24/2008	Second draft
Grant Hill	10/11/2008	Third draft following call for comments

Introduction

In this document, requirements are called out for a suite of software capabilities which will facilitate an instrument preventive and predictive maintenance program. The requirements are subdivided into a number of categories to help guide our planning this program.

1. Database of tasks

- 1.1 Software for the instrument PM program will include a database containing task information.
- 1.2 The database will support the linking or storage of procedural information, pictures, drawings and vendor contacts for each task.
- 1.3 The database will be searchable by instrument, task type, recurrence frequency, assignee, overdue status, priority, and threat finding.
- 1.4 The database structure will be sufficiently general that it can be extended for future instruments.
- 1.5 The database structure will be sufficiently general that it is able to include at some point, systems not currently considered part of the instruments; eg. guiders, CCR's, and dome lamps.
- 1.6 It will be easy to display, add, remove, or modify task information in the PM task database.
- 1.7 The languages and tools used to build the database will be chosen with regard to their popularity, familiarity, and prospects for continued use within Keck.
- 1.8 Proprietary or licensed database technology is to be avoided.
- 1.9 The database will be backed up regularly (daily if feasible but at least weekly).

2. Scheduling

- 2.1 Software will automatically email work orders to appropriate staff when maintenance tasks are due for completion.
- 2.2 Scheduling software will allow the linking of tasks to related day log or night log tickets. These links will operate in both directions such that a ticket can trigger a work order earlier than normal scheduling would have otherwise and performance of the work order can represent a response to the ticket.
- 2.3 Work orders will contain links to pictures, procedures and documentation necessary for completion of tasks.



- 2.4 Work orders will contain links to enable the notes, data, and measurements that result from performing the task to be archived properly when this is appropriate. In some cases a minimum response will be an acknowledgement that the task was completed.
- 2.5 When work orders are not responded to, automatic reminders should be sent out after some waiting period. The software will allow different waiting periods to be assigned depending on task. The list of recipients for reminders should not necessarily have to be identical to that of the original work order.
- 2.6 After some additional waiting period, if a task reminder has not resulted in action the software should produce a warning email. The recipient list for this email should not have to be identical to either that of the original work order, nor the reminder.
- 2.7 Waiting periods and recipient lists should all be easily modified for each PM work order, reminder and warning.
- 2.8 Global recipient lists and waiting periods should be definable.
- 2.9 The scheduling software should allow “across the board” delays of work order issuances in the event of extended periods of restricted summit access, personnel unavailability, breakdowns, or other events.
- 2.10 The scheduling software will check the telescope schedule to avoid requests for work on instruments on sky.
- 2.11 The scheduling software will check the people calendar to avoid requests for work by people on leave.
- 2.12 All email work orders, reminders and warnings should be formatted so as to facilitate printing a hardcopy for use at the work site.
- 2.13 The scheduling software will have the ability to automatically create look forward summaries indicating what tasks are coming up in the next week or month and look backward summaries detailing work done or overdue in the past week or month.

3. Archiving and record keeping

- 3.1 An archive will exist to store the products of PM task performance.
- 3.2 The archive should be capable of storing a wide variety of PM performance products. These will include but will not be limited to binary (done/not done), numeric, observational notes, and measurements, pictures, and FITS files.
- 3.3 The archive will be searchable using a wide variety of criteria such as instrument, task type, date range, assignee, overdue status, and threat finding.
- 3.4 The archive will support the retrieval, display and analysis of PM performance products.
- 3.5 The archive will be extendable to future instruments.
- 3.6 The archive will be extendable to systems not currently considered part of the instruments such as guiders, CCR's , and dome lamps.
- 3.7 It should be simple for the archive to ingest new products.
- 3.8 The archive software will be chosen with regard to its popularity, familiarity, and prospects for continued use within Keck.
- 3.9 The archive will be backed up regularly (daily if feasible but at least weekly).



4. Analysis capabilities

- 4.1 Web based, interactive tools are desirable for the manipulation, visualization and analysis of the results and products of PM task performance.
- 4.2 Visualization capabilities will include plotting selected quantities versus time, versus each other, over-plotting multiple quantities, histograms, and pie charts.
- 4.3 It will be possible to pan and zoom on 2-dimensional graphs by specifying min-max ranges.
- 4.4 It will be possible to specify tick frequency on graphs to aid interpretation and avoid strange units.
- 4.5 Analysis capabilities will include the ability to calculate simple statistical quantities like averages, trends (slopes), and standard deviations about means.
- 4.6 Scripts will exist to reduce and analyze the products of the performance monitoring data gathering scripts described in section 6 below. Whenever feasible, these scripts will run in an automated fashion. Whenever feasible, they will compare their results to some expectation or historical trend and alert e-mail recipients if a potential problem exists.
- 4.7 Whenever appropriate, scripts will exist to mine and analyze data from science data taken by observers. This will include, but not be limited to examples such as examination of CCD over-scan regions, observations of standard stars, and instrument parameters written to log files.

5. User interfaces

- 5.1 The GUI's for the instrument PM program will be interactive, should be web based, and if web based, compatible with browsers in use at Keck on both the PC and UNIX networks.
- 5.2 The analysis GUI's will be capable of producing hard copy, saving plots to files, CSV output, and in general producing output that can be included in reports, posted on web pages, and included in Excel spreadsheets.
- 5.3 A GUI will allow users to view, add, remove, or modify entries in the database of PM tasks. The mechanism for entering notes will offer the ability read text from a file.
- 5.4 A GUI will allow users to show status of PM tasks. It will allow one to display one's choice of one or more of categorized tasks. For example, one should be able to display all pending tasks, all those assigned to a particular person, all those completed within a certain time frame or for a certain instrument.
- 5.5 A GUI will allow users to modify task assignment lists, assignee contact lists, recurrence frequency, due dates, and waiting period lengths in the scheduling software.
- 5.6 A GUI will facilitate the ingestion of new results into the archive of PM program products.
- 5.7 As much as possible, GUIs will make use of drop-down menus, radio buttons, etc. so as to provide a similar look and feel to day-log entry pages.



6. Instrument control scripts

- 6.1 Whenever appropriate, unix shell scripts will allow instruments to be configured for preventative maintenance tasks that involve inspections or actions such as lubrications, realignments, and adjustments.
- 6.2 Whenever appropriate, unix shell scripts will allow instruments to be configured and run to obtain predictive maintenance data such as best focus positions, motor torques, encoder signal quality, or temperatures.
- 6.3 Whenever appropriate, unix shell scripts will allow the gathering of performance monitoring data to be automated. Such data will include, but not be limited to, data for the calculation of bias levels, gains, dark currents, nonlinearity, throughput, and residual charge.
- 6.4 The capability will exist to launch the scripts described above, from the work orders requesting them, when this is deemed desirable, appropriate, and safe.
- 6.5 The capability will exist to run the scripts described above, as part of cron jobs, when this is deemed desirable, appropriate, and safe.
- 6.6 The scripts will be backed up regularly (daily if feasible, but at least weekly).
- 6.7 A mechanism will be provided to script developers so that they can store the results of each test directly into database tables.
- 6.8 A mechanism will be provided to script developers so that they can read test configuration parameters (e.g. the directory path to be used as a scratch area for writing intermediate files) directly from database tables.
- 6.9 The database schema will be designed so that new monitors can be added without requiring programmers to make fundamental changes to the database (e.g. add new tables).
- 6.10 A file naming protocol and directory hierarchy will be designed to scripts to eventually number in the hundreds without becoming unmanageable.