

MONSOON Software Review

Appendices

- I. Methods to Achieve Requirements.
- II. Interface Control documents.
 - A. ICD 4.0 GPX Command, Status and Data Interface.
 - B. ICD 5.0 PPX Command, Status and Data Interface.
 - C. ICD 6.0 Generic DHE Command Interface.
 - D. ICD 6.1 MONSOON PAN to DHE Interface.
- III. Original Project Documents
 - A. MONSOON OCCD
 - B. MONSOON FPRD

Methods for Achieving MONSOON Software Requirements.

Science Generated Requirements
Detector Development Requirements
System Generated Requirements
Software Development Requirements

Science Software Requirements

- Support detector safe operations.
 - Use “Standard Parameter Sets”.
 - Parameter range restrictions in science operations.
 - Only Limited parameter tuning permitted.
 - Bias power off until detector voltages confirmed correct.
 - Detector engineer determines “safe” operation.
- Support detector limited performance.
 - Use high-end giga-Hz CPU’s.
 - No restrictions on speed built into software.
 - Allow overlap of execution to greatest extent possible.

Science Software Requirements (cont)

- Support both IR and OUV detector systems.
 - Per detector runtime command configuration.
 - Runtime pre-processing algorithm selection.
 - Base system code non-specific.
 - User interface left to Observatory/Instrument staff.
- Extensible to “Very Large Focal Planes”.
 - Individual PAN’s know only about their FP section.
 - Final image assembly left to DHS.
 - MSL knows how to deliver commands to FP sections.

Science Software Requirements (cont)

- Able to treat mosaics as single focal plane.
 - User view is of an image server.
 - MSL deals with details of multiple PAN's.
 - Data sent to DHS by individual PAN's for reassembly.
- Support science observation by 'named' modes.
 - Mode database converts names to Parameter settings.
 - MSL knows how to send parameter settings to PAN's.
 - Instrument/detector scientists/engineers determine Modes.
 - User can add features to base modes within restrictions.
 - User modes savable in database.

Science Software Requirements (cont)

- Provide for efficient science operation support.
 - Allow overlap of execution to greatest extent possible.
 - Provide Modes to support Focus, Calibration frames, etc.
 - Provide fast setup and configuration.
- Support high observing efficiency.
 - Allow overlap of execution to greatest extent possible.
 - Provide Modes to support Focus, Calibration frames, etc.
 - Provide fast setup and configuration.
- Support existing and new observing paradigms.
 - Connections permit Remote observing, Queue observing,
 - Published Interfaces allow automation of observations.
 - Facilities for adding functionality allow new modes.

Science Software Requirements (cont)

- Provide ROI support for readout speed up.
 - Single ROI to reduce readout time.
 - ROI spans chips in restricted ways.
 - Allowed alternatives determined by mosaic layout.
- Provide ROI support for data compression.
 - Readout time not reduced all pixels readout.
 - Multiple ROI's allowed.
 - Only data within the ROI's sent to DHS.
- Support “technical” imaging. (Guiders, etc.).
 - Runtime configuration allows customization.
 - Detector library permits unique handling for purpose.

Science Software Requirements (cont)

- Array configuration by ‘standard parameter set’.
 - Detector setup by named Mode.
 - Parameter set database converts names to parameter settings.
 - Provision for expansion of Database by Users.
- Parameter sets determined in the Detector Lab.
 - Initial database created by detector engineer.
 - Instrument scientist can tune modes for instrument.
 - User has limited ability to modify base operating mode.
- Provide limited tuning of detector performance.
 - User can modify restricted range & class of parameters.

Science Software Requirements (cont)

- Provide limited ‘on-the-fly’ reconfiguration.
- Allow addition of new processing algorithms.
 - Processing algorithm can be restarted with new algorithm.
 - New data taking algorithms can be loaded.
 - New detector waveforms can be loaded as needed.
- Provide for an instrument calibration mode.
 - Instrument Client can run MSL/PAN’s automatically.
 - Password restriction allows access to restricted parameters.
 - Engineering console can be used to tune parameters.

Detector Development Requirements

- Support detector characterization operations.
 - ppx interface provides Low-level control of parameters.
 - Interface allows automated control of process.
 - “on the fly” reconfiguration allows new algorithm development.
 - Library & process APIs allow convenient development.
- Support detector research and development.
 - Runtime configuration allows new detector types.
 - “on the fly” reconfiguration allows new operation modes.
- Support hardware development & debugging.
 - Ppx interface allows Low-level hardware command support.

System Software Requirements

- Support efficient boot-up and initialization.
 - Each PAN boots up independently.
 - Boot-up only depends on local information.
 - Runtime configuration sent to all PAN's.
 - Communication mechanism “universal” & easy to use.
- Start-up & initialization without intervention.
 - Startup scripts keyed to Instrument/system.
 - MSL does Startup time checking to insure correct operation.
 - Crashed/down nodes restarted automatically.
 - Start-up monitored by logging mechanism.

System Software Requirements

- Support system operations logging.
 - Error detection and recovery logging.
 - Command sequence playback.
 - Status display of current state saved with exposure.
- Support connection security.
 - Multiple connections allowed.
 - Connection security enforced (firewall, IP restrictions).
 - On-Telescope connection priority observed.
 - Engineering level commands password protected.

System Software Requirements

- Support convenient error detection/recovery.
 - Routines track and inherit status.
 - First error is reported and correction is attempted.
 - Errors reported and corrected automatically if possible.
 - Error help included in report.
- Support remote diagnosis, debug & operation.
 - Multiple connections permitted.
 - Engineering connection password protected.
 - Test facilities built into system at several levels.

System Software Requirements (cont)

- System to include simulation capabilities.
 - Hardware libraries provide simple simulation.
 - Simulation automatic in absence of Hardware.
- Help available for commands/parameters.
 - Help provided by Runtime Configuration process
 - Command help available for every command.
 - Parameter help available for settable parameters.
- System layered to allow maximum reuse.
 - General Purpose & Generic Interface libraries reusable.
 - Top level process code reusable.
 - Framework of Hardware Specific libraries reusable.

System Software Requirements (cont)

- Pixel data processing chain re-configurable.
 - Multiple processing algorithms supported in panProcAlg.
 - Multiple panProcAlg versions supportable.
 - Process interconnects will be well defined and published.
- Support efficient configuration for new systems.
 - Engineers define hardware with spread sheet.
 - Runtime configuration derived from spread sheet.
- Configurable w/o re-compile of base code.
- Features added without rebuilding system.
 - Hardware specific libraries compiled as shared libraries.
 - New configurations read at runtime.

System Software Requirements (cont)

- Software to use well-defined interfaces.
- Interfaces documented and published.
- Use GPX Interface to outside world.
- Documentation maintained with code base.
- Documentation in standardized format.
 - Library API created by “make” process and published
 - Process interface being documented and published.
 - GPX or PPX interface used as interfaces to Clients.
 - API’s, ICD’s & Process interface externally available.\
 - Web publication planned.

System Software Requirements (cont)

- Support “Package-like” Installation.
 - Planned.
- Support verification & removal of old versions.
 - Planned.
- Develop Source code Maintenance Manual.
 - Will be developed from source code API creator.
 - Manual will include library and Process descriptions.

Development Software Requirements

- All Software to be Open Source.
 - Code available on externally visible disk.
 - WEB obtainable source code planned
- Use widely available software technologies.
 - Sockets, Semaphores, Shared Memory.
 - ANSI standard 'C' compiler.
 - 'POSIX' compliant standard Libraries.
- Use free tools to the greatest extent possible.
 - 'C', LINUX, TCL/TK.
 - Firmware using XLINIX WEBPACK

Development Software Requirements

- Support multi-site distributed development.
 - CVS Tree on externally visible disk
 - Individual modules obtainable through CVS. (NI)
- Use Source code version control (CVS).
 - CVS Tree built and in use at NOAO.
 - CVS Tree available on externally visible disk.
- Testing and verification built into development.
 - Testing plan for Libraries
 - PAN process test-bed planned
 - dheSim complete and operating at rudimentary level.

