Gen2 Technical Overview

Eric Jeschke

eric@naoj.org

with
Inagaki T., Streeper S., Tomono D., Terada H., Takami H.
Outline

- Overview
- Goals
- Software Architecture
  - Phase 1
  - Example: Monitoring
  - Phase 2
  - Compatibility Mode
  - Phase 3
- Hardware
- Roadmap
- Questions
Goals for Gen2 OCS (Observers & Obs Support Staff)

- Improved language & tools for writing observation tasks
- Improved graphical user interfaces for observation
- Clear, simple and transparent monitoring of command execution
  - WHAT is happening, WHERE and WHEN
- Good documentation
- Portable version for training, simulation & instrument development
- Reliable operation & simplified troubleshooting
- ??? (your opinion here)
Goals for Gen2 OCS
(Developers & Support Staff)

- Flexible software design for future modifications and astronomer workflows
- Automated, comprehensive regression tests
- Software version control and simple upgrade/downgrade procedure
- Simple, centralized logging and monitoring
- Good fault-tolerance (resistance to failure) via graceful failover or simple reconfiguration
- Simple start up, shut down, troubleshooting
- Simple maintenance
Gen2 Software Diagram

Phase I

Simulators
- Status Simulator
- SIMCAM
- TSCCAM
- AgSim
- STARS Simulator

Gen2 Back Ends (Servers)
- Legacy Status Interface
- Status
- Legacy Instrument Interface
- Telescope Interface
- Guider Interface
- Legacy STARS Interface
- Session Manager
- Frame Server
- Support Modules
- Boot Manager

Subaru Remote Objects Infrastructure
- Manager Service
- Name Service

(Task Manager
- Gen2 Tasks

Monitor

(No SOSS code)
Example: Monitoring

- Need to know what is happening: what commands, what subsystems
- Lesson from current OCS: need both “birds eye view” and detail (if wanted)
- Architecture of Gen2 is designed to facilitate centralized reporting via the Monitor (a kind of shared, high-bandwidth bulletin board)
Gen2 Monitor

Eric Jeschke

Observers and Obs Support Staff

IntegGUI

OCS Displays

Task Manager

Gen2 Tasks

Telescope Interface

Telescope

Instrument Interface

Instrument

Monitor
Gen2 Monitor
Eric Jeschke

Observers and Obs Support Staff

IntegGUI

OCS Displays

Task Manager

Telescope Interface

Gen2 Tasks

Instrument Interface

Monitor

Telescope

Instrument
Gen2 Monitor
Eric Jeschke

Observers and Obs Support Staff

IntegGUI

OCS Displays

Task Manager

Gen2 Tasks

Monitor

Telescope Interface

Telescope

ack

Instrument Interface

Instrument

ack

ack
Gen2 Monitor
Eric Jeschke

Observers and Obs Support Staff

IntegGUI

OCS Displays

Task Manager

Gen2 Tasks

Monitor

Telescope Interface

Telescope

Instrument Interface

Instrument
SOSS Compatibility Mode

- Ability to run legacy observations
  - Support execution of most .sk/.ope files
  - Support some legacy GUIs (e.g. IntegGUI, TelStat, VGW, QDAS, Skymonitor)
- Allows Gen2 use with minimal retraining
- Allows time to develop new user interfaces
Gen2 Software Diagram

Phase III

Gen2 Back Ends (Servers)
- Status Simulator
- SIMCAM
- TSCCAM
- AgSim
- STARS Simulator

Gen2 Tasks
- Skeleton File Interp
- Frame Server
- Support Modules
- Boot Manager
- Session Manager

Gen2/SoSS Compatibility Mode

Simulators

Gen2 Front Ends (Clients)
- New GUIs (prototypes or NY1)
- Quick Look Pipeline
- General FITS Viewer
- IRAF, etc.

Data Analysis

Legacy Status Interface
- Status
- Monitor
- DB

Legacy Instrument Interface
- Telescope Interface
- Guider Interface
- Legacy STARS Interface
- Task Manager
- Gen2 Tasks

Subaru Remote Objects Infrastructure
- Manager Service
- Name Service

Old GUIs (ported from SoSS & adapted)
- IntegGUI
- Misc QDAS
- FITS Viewer
- Sky Monitor
- VGW Subsystem
Hardware

- High-Availability cluster design
  - Redundancy and fault-tolerance
- Commodity hardware (when possible)
- Simple maintenance and replacement
- Full remote manageability
- Extensive monitoring of fans, temp, disks, memory, CPU, network, etc.
Hardware Redundancy

- No specialized nodes: any node can do any job
- Redundant components to provide fault-tolerance and graceful hardware failover
  - All disks in RAID configurations
  - Dual power supplies
  - Dual network ports in bonded configuration
  - Multiple fans
  - Multiple CPUs
Simple Maintenance and Repair

- All disks are front-accessible hot-swap
- CPUs are generic x86 rack mount units, easily replaceable
- System software (OS, etc) is “cloned” to a new system in minutes
Full Remote Managability

- All hardware in Gen2 cluster can be managed over the network from Hilo base
- IP-enabled KVM (keyboard/video/mouse)
- IP-enabled power distribution
- IP-enabled UPSes
- IP-enabled switches
- e.g. turn off power, reboot, access console, etc.
Hardware Monitoring

- Hardware is monitored continuously
- Alerts and statistics easily seen via web interface
Gen2 Development Timeline

Eric Jeschke

- Gen2 project start: 2005/06
- 1st Eng: 2006
- 2nd Eng: 2007
- SOSS Compatibility Mode: 2008
- 3rd Eng: 2009
- 4th Eng
- Begin Native Front Ends: 2009
- 5th Eng
- Stop development

Gen2 Native Mode
Gen2 Rollout: 2009

• To smooth adoption of Gen2 we must manage the risk of using new software
• → Make it simple to “fall back” to using SOSS on any given legacy-style observation
• Develop a simple, quick switch procedure between Gen2 and SOSS observation
• Share frame id allocation and ensure accessibility of data files from both sides
• Begin with risk share, staff obs, new instruments
Questions?

Thank You!

eric@naoj.org