MODS Instrument Software

Richard Pogge

The Ohio State University

Department of Astronomy

MODS Progress Review – 2004 April 22

MODS Instrument Software

MODS Software Requirements

Software Design Approach

- Coding & Documentation Standards

MODS Software Components

Instrument Control System

Software Progress to Date

Software Development Plan

Staffing Issues

MODS Software Requirements

Integrate all MODS functions:

- Instrument configuration (mechanisms)
- Data-acquisition (CCDs) & Data Handling
- Autonomous subsystems (AGW, IMCS, etc).

Provide LBT Observatory Interfaces:

- Telescope Interface (IIF library)
- AGW System (XY stage carries AGW cameras)

Provide User Interfaces:

- Observer interface (local & remote)
- Engineering Interfaces
- Observing Preparation & Planning Tools

Software Design Approach

Common system applied to all instruments

- Builds on our 15-year instrumentation heritage
- Open, modular architecture
- Adapts and evolves for each instrument

Aggressive Prototyping:

- MODS elements already in deployed instruments.
- Develop & test mechanism software in parallel with mechanism assembly & testing.

Seek solutions of appropriate scale

- Keep things in perspective…

MODS Coding Standards

Open Source/Open Standards

- Use established standards (ANSI C/C++, POSIX).
- All source code will be public domain (GPL).
- Use common public domain packages & utilities.

Public-domain packages:

- GNU readline/history library for CLI shells.
- X11/Qt package for GUI development.
- pthreads library for multithreading/concurrency.
- cfitsio libraries for FITS support.
- SAOImage DS9 & XPA for image display.

MODS Software Documentation

Software Documentation:

- Doxygen for code-level software documentation.
- Online (HTML) and Offline (PDF via LaTeX) documents.

User Documentation:

- System Engineering Manuals (HTML+PDF)
- Observer's Manual (HTML+PDF)
- Observer's quick-reference (HTML+PDF)
- Online documents with indexing & searching.

MODS Software Components

Instrument Control System

- Low-level interfaces to microstep drives & sensors
- High-level programs for integrated mechanism control.
- Independent modules for each major mechanism group.

Detector Control System

- Array configuration and readout.
- Exposure timing and control.
- Deliver images in standard FITS format.

MODS Software Components

Data Handling System

- Image storage (RAID arrays)
- Image display (SAOimage ds9)
- Data logging

Observatory Interfaces

- IIF interface for MODS/TCS communication
- Interface for the AGW stage and LBT AGW system

User Interfaces

- MODS User Interface (MUI): observing console
- MODS Engineering/Support Interface (MESI)

Supporting Software Components

Observing Planning Tools:

- Skyview-style observing planning tools.
- "Phase 2" Observation Template preparation tools.
- Support for Remote/Queue-Service modes.

Slit Mask Design & Manufacture:

- Packages to distribute to partners for slit mask design in advance of a run.
- Mask-making software (laser cutting machine).
- Bar code generation and mask inventory tracking.
- Create in collaboration with LUCIFER group.

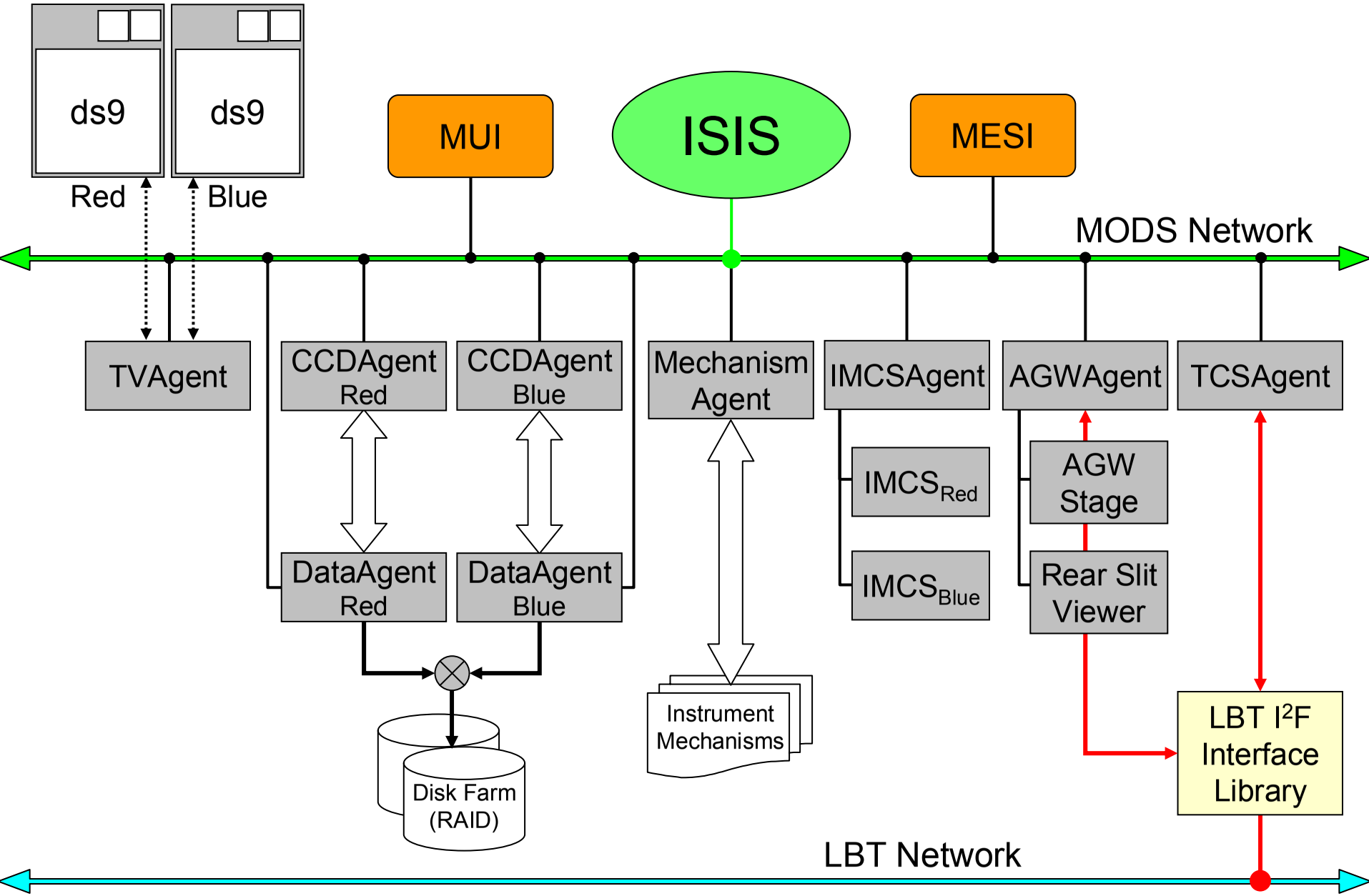
Instrument Control System Model

Client/Server System

- Interprocess communications using a text-based messaging protocol (IMPv2).
- Isolate real-time-critical or "blocking" tasks in dedicated systems linked by a server.

Open Modular Architecture

- Modules communicate and cooperate, but are not deeply interdependent.
- User interfaces are just one disposable module (GUI, CLI, whatever).
- Common client library – easy to write new apps.


ds9
ds9
Red
Blue
MUI
ISIS
MESI
MODS Network
TVAgent
CCDAgent Red
CCDAgent Blue
Mechanism Agent
IMCSAgent
AGWAgent
TCSAgent
IMCS Red
IMCS Blue
AGW Stage
Rear Slit Viewer
DataAgent Red
DataAgent Blue
Instrument Mechanisms
Disk Farm (RAID)
LBT I2F Interface Library
LBT Network

ISIS

Integrated Science Instrument Server

Message-passing server & interface multiplexer

- Uses the IMPv2 messaging protocol
- Network socket communications (UDP sockets)
- Serial (RS-232) capability.
- Interactive command-line interface

Services:

- Passes IMPv2 messages between clients
- Processes broadcast messages for all clients
- Provides runtime logging of all system telemetry
- Handles “executive” server commands

ISIS – Deployed Systems

ANDICAM at CTIO 1.3m Telescope

- 2-channel IR+CCD camera redeployed in Jan `03
- Hybrid of ISIS and existing MS-DOS systems
- 1 year of successful science operations

Y4KCam at CTIO 1-m Telescope

- Full ISIS system with CCD and TCS agents
- Deployed in March `04, operations start May `04

OSIRIS at SOAR 4.2m Telescope

- Installed in March `04 (ISIS+MS-DOS hybrid)
- Commissioning observations begins in May `04.

MODS Software Progress

Server & client API libraries:

- ISIS server written and deployed.
- ISISClient library written and deployed.
- Messaging & Transport infrastructure is in place.

Instrument Control System (ICS):

- Test and evaluation of microstep drive systems.
- Deployed systems using MODS prototype software.
- Developing low-level (microstep drive API) and high-level mechanism control agents.

ICS Requirements

Concurrent configuration of mechanisms where it is safe & sensible.

High degree of modularity:

- Same software/hardware used on/off instrument
- Heavy reuse/inheritance of common components

Adaptive runtime logging of all operations

- Concise, verbose, and ultra-verbose modes.

Mutual exclusion handlers for mechanisms that have to serve 2 masters:

- AGW Stage shared by MODS and LBTO AGW.

Mechanism Groups & Topologies

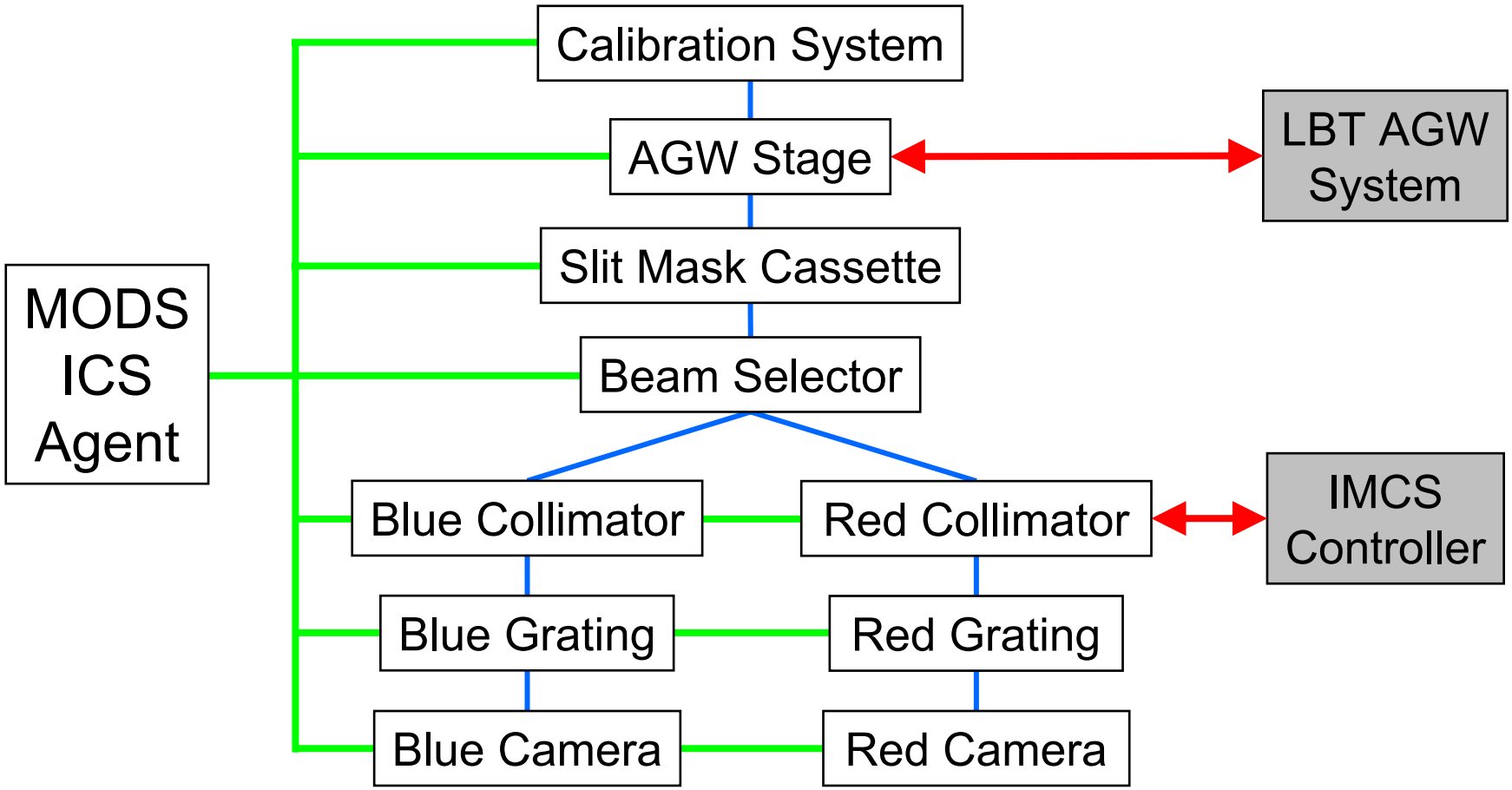
Each MODS has 32 mechanisms, but…

- Only 7 independent mechanism groups
- Only 3 basic drive topologies:
  - Rotary-indexed drives (e.g., filter wheels)
  - Linear-indexed drives (e.g., slit mask select)
  - Linear-continuous drives (e.g., focus motors)

All mechanisms are instances of one of these three topologies

- Need to only develop 3 basic "classes" of microstep drive interfaces.

Mechanism Group Schematic

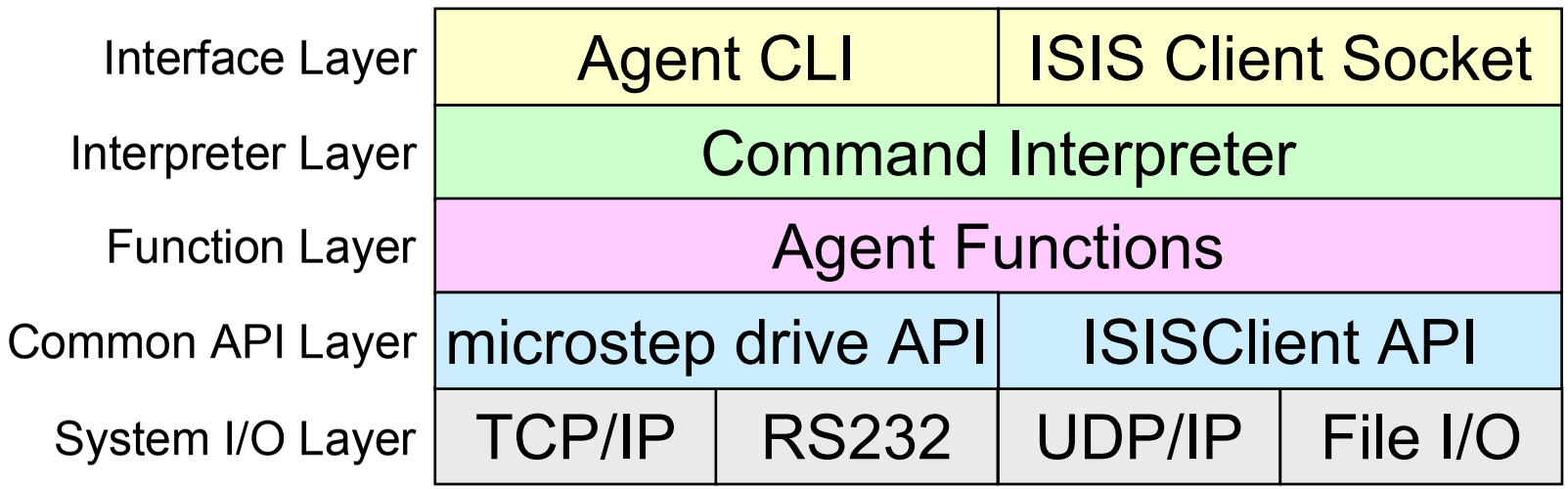


Mechanism Agent Architecture

All individual mechanism agents share a common architecture and look-&-feel.

Built on common API libraries.

Can run standalone or as ISIS clients.



Mechanism Agent Development

- 2 filter wheels deployed at CTIO & MDM using a prototype rotary-indexed mechanism controller.
- MODS filter wheel uses v1.0 of microstep drive API.
- Linear-slider mechanism controller will be developed using the 3-axis collimator actuator system, including multithreading for concurrent motions.
- Linear-indexed mechanism controller will be developed using the grating select mechanism for prototyping.
- Blue Camera System (filter wheel, camfocus, shutter) the first major system with mixed-motion controls.
- Bring other systems online as they are assembled.

The Next Steps

Instrument Control System:

- Development is running in parallel with instrument mechanism subsystem assembly & testing.

Detector Control System:

- Pending the final choice of the detector system…
- Prototypes for main options already largely exist.

Data Handling System:

- Develop requirements and specify components.
- Details depend on the choice of detector system…

The Next Steps (cont'd)

Observatory Interfaces:

- Have IIF and LBT simulators, will create an LBT "tcsagent" client this summer for testing.
- AGW camera stage interface will be developed in collaboration with the LBT AGW effort.

User Interface Development:

- MESI will emerge during instrument assembly – required for lab acceptance testing.
- MUI developed & deployed during lab performance testing.

Staffing Issues

Rick Pogge

- Overall management & software design
- Test and evaluation
- User Interface & LBT Interfaces

Jerry Mason

- computer and data-handling hardware
- data-handling & detector software
- general software tasks

Need a new full-time programmer

- Will take on main coding tasks