

The Mosaic Data Capture Agent

Doug Tody and Francisco G. Valdes

IRAF Group, NOAO¹, PO Box 26732, Tucson, AZ 85726

Abstract. The Mosaic Data Capture Agent (DCA) plays a central role in connecting the raw data stream from the NOAO CCD Mosaic controller system to the rest of the data handling system (DHS). It is responsible for assembling CCD data from multiple readouts into a multiextension FITS disk file, for supplying a distributed shared image to real-time elements of the DHS such as the Real-Time Display (RTD), and for feeding data to the IRAF-based data reduction agent (DRA).

1. Overview

A brief design of the data handling system (DHS) for the NOAO CCD Mosaic Camera² was presented earlier (Tody, 1997). It is based on a message bus architecture (Tody, 1998) that connects the various components of the DHS. One such component is the *data capture agent* (DCA). The DCA receives streams of readout data from the acquisition system and builds a data object which is shared by other components of the DHS. The image data is stored on disk as a multiextension FITS file (MEF) (see Valdes 1997b).

The DCA is a general purpose network-based data service, using the message bus as the network interface. It receives request messages of various types and, through an event loop, dispatches each request to the appropriate event handler. The request message types currently implemented in the DCA are for control, setting or querying server parameters, readout status, data format configuration, and writing header and pixel data. In addition to servicing client requests, the DCA can broadcast messages to inform clients of the status of the DCA during readout.

The major subsystems of the DCA at present are the message bus interface, an event loop and event handlers for processing message requests, a distributed shared object implementation, a keyword database system, the TCL interpreter, and a TCL-based keyword translation module.

¹National Optical Astronomy Observatories, operated by the Association of Universities for Research in Astronomy, Inc. (AURA) under cooperative agreement with the National Science Foundation.

²<http://www.noao.edu/kpno/mosaic/>

2. DCA User Interface

The DCA may be started by issuing a command at the host level, or invoked as a service via the message bus. This can occur during login to an observing account, using a window manager menu, or by interactively issuing a user command. The command invocation can include DCA server parameters which can also be set or reset after invocation via messages.

The DCA automatically connects to the message bus when it is started. The runtime operation of the DCA is monitored and controlled by clients via the message bus. In the Mosaic system there are two clients; the data feed client (DFC) and the DCA console client (DCAGUI). In general any number of clients can access the DCA. Multiple data feed clients or GUIs can be active simultaneously.

The DCA console client has a graphical user interface based on the IRAF Widget Server (Tody, 1996). It can send control and parameter messages to the DCA and act on messages broadcast by the DCA, e.g., to display the readout status. Currently the DCAGUI executes an *autodisplay* command for real time display during readout (see Valdes, 1998) and a post-processing command for logging, archiving or other operations once the readout ends.

3. The Readout Sequence

An observation readout is driven by a sequence of DCA message requests. A readout sequence is initiated by the DFC with a message that includes a unique sequence number. Once a readout has started, each subsequent message associated with that readout must be tagged with the sequence number for that readout. Multiple readouts can be simultaneously in progress. Each readout sequence has a separate context identified by its sequence number.

When a readout sequence is initiated the DCA creates a new image-class *distributed shared object* (DSO). The DFC passes information about the size and structure of the object (such as the number of image extensions), allowing the DCA to configure the DSO. At this point the DFC can begin sending pixel and header messages. At image creation time the DCA broadcasts a message so that clients like the DCAGUI can access it if desired. Currently the DCAGUI uses this to initiate an interim real-time display.

When the readout is completed the DCA executes a *keyword translation module* (KTM), an externally supplied, interpreted TCL script which converts detector specific information into standard header keywords. After the KTM finishes the DCA and DSO format the output keywords, write them to the image headers, and generation of the new image DSO is complete. The DCA broadcasts a message when the DSO is complete which can be used to trigger post-processing of the data.

3.1. Pixel Messages

The pixel data messages contain blocks of raw, unprocessed detector pixel data organized into one or more *streams*, one for each CCD amplifier. Each stream directs pixels to a region of an output image extension. This structure allows the data block to simultaneously contain data for several different regions and

the data can be arbitrarily interleaved, encoded, flipped, or aliased. Each block of data is processed asynchronously but the client can send a synchronization request periodically to check the output status.

3.2. Keyword Messages

The DCA receives header information via the message bus. This information consists of blocks of keywords organized into named, detector specific *keyword groups*. The keywords are stored in keyword databases (an internal random access data structure), one per keyword group. The set of keyword group names is arbitrary. Some examples for the NOAO Mosaic are ICS for instrument control system, TELESCOPE for telescope, and ACEB n for Arcon controller information for controller n .

The Arcon controller system for the NOAO Mosaic consists of a set of Arcon controllers each of which can readout one or more amplifiers. The current system has four controllers each reading out two CCDs using one amplifier per CCD. Thus there can be controller information for the whole system, for each controller, and for each amplifier readout.

A keyword database library handles creation and maintenance of the keyword database. Note that keyword information does not necessarily have to come only from the DFC, the current mode of operation for the NOAO Mosaic. Other schemes are possible.

4. Keyword Translation Module

The keyword translation module (KTM) is a TCL script called by the DCA at the end of a readout, once all raw header information has been received. The purpose of the KTM is to create the keywords for the global and extension headers. The KTM is passed a list of input keyword database descriptors and it returns a list of output keyword database descriptors, one for each header. The DCA TCL interpreter provides special TCL commands for manipulating (creating, accessing, searching, etc.) the keyword databases. When the KTM finishes the DCA, via the DSO, writes the keywords in the returned output keyword databases to the output MEF FITS file.

The KTM performs a variety of transformations on the input keywords. A keyword can be copied verbatim if no change is desired. The keyword name or comment may be changed without changing the value. New keywords can be added with static information and default values may be supplied for missing keywords. The KTM can compute new keywords and values from input keywords. Identical keywords in each extension may be merged into a single keyword in the global header. The KTM can detect incorrect or missing keywords and print warnings or errors.

Two examples from the keyword translation module for the NOAO CCD Mosaic follow.

1. The data acquisition system provides the keywords DATE-OBS and UT-SHUT giving the UT observation date in the old FITS date format (dd/mm/yy) and the UT of the shutter opening. The KTM converts these

to TIME-OBS, MJD-OBS, OBSID, and the new Y2K-compliant FITS date format.

2. The KTM determines on which telescope the Mosaic Camera is being used and writes the WCS scale, orientation, and distortion information previously derived from astrometry calibrations for those telescopes. The coordinate reference point keywords, CRVAL1 and CRVAL2, are computed from the telescope right ascension and declination keywords.

5. Distributed Shared Objects and Distributed Shared Images

An important class of message bus component is the distributed shared object (DSO). DSOs allow data objects to be concurrently accessed by multiple clients. The DSO provides methods for accessing and manipulating the data object and locking facilities to ensure data consistency. DSOs are distributed, meaning that clients can be on any host or processor connected to the message bus. In the case of the Mosaic DHS, the principal DSO is the distributed shared image which is used for data capture, to drive the real-time display, and for quick look interaction from within IRAF. The distributed shared image uses shared memory for efficient concurrent access to the pixel data, and messaging to inform clients of changes to the image.

The current version of the DCA does not implement the full distributed shared image object. Instead it implements a mapped image file. The image being created appears as a valid FITS multiextension file immediately after the configuration information is received from the DFC. This allows applications to examine the file while the readout is in progress. An example of this is the interim display routine that loads a display server frame buffer as the pixel data is recorded, giving a simple real-time display capability.

6. Current Status and Future Developments

A version of the Data Capture Agent is in production use at two telescopes on Kitt Peak with the NOAO CCD Mosaic Camera. The flexibility of the message bus architecture is used to provide two modes of operation. The standard mode uses two machines connected by fast Ethernet. One machine supports the observing user interface and the interface to the controller system. A DFC running on this host writes to the message bus and the DCA runs on another machine (with a different OS) where the exposures are written to disk and the quick look interaction and any post-processing are done. The second mode is a fallback in case the second computer fails. The DCA can simply be run on the same machine as the UI and controller interface.

Future developments will complete the distributed shared object and console client and add other clients such as the real-time display (Tody, 1997) and data reduction agent (Valdes 1997a).

The design of the DCA (and the whole DHS based on the message bus architecture) is open, flexible, and efficient. It can be used with many data acquisition systems at a variety of observatories. All that is needed is to write a data feed client to connect to the message bus and a keyword translation module

appropriate for the data. Currently the lower level messaging system is based on the Parallel Virtual Machine (PVM) library but this can be replaced in the future with other messaging systems such as CORBA.

References

- Tody, D. 1996, in ASP Conf. Ser., Vol. 101, *Astronomical Data Analysis Software and Systems V*, ed. George H. Jacoby & Jeannette Barnes (San Francisco: ASP), 89
- Tody, D. 1997, in ASP Conf. Ser., Vol. 125, *Astronomical Data Analysis Software and Systems VI*, ed. Gareth Hunt & H. E. Payne (San Francisco: ASP), 451
- Tody, D. 1998, this volume
- Valdes, F. 1997a, in ASP Conf. Ser., Vol. 125, *Astronomical Data Analysis Software and Systems VI*, ed. Gareth Hunt & H. E. Payne (San Francisco: ASP), 455
- Valdes, F. 1997b, in ASP Conf. Ser., Vol. 125, *Astronomical Data Analysis Software and Systems VI*, ed. Gareth Hunt & H. E. Payne (San Francisco: ASP), 459
- Valdes, F. 1998, this volume