Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

The CARMA Monitor System (CAM) - Transforming Cyclically Collected Telemetry into a Linear Stream

N. S. Amarnath[1], Stephen L. Scott[2], James Colby Kraybill[3], Andrew D. Beard[2], Paul Daniel[2], Chul Gwon[1], Rick Hobbs[2], Erik Leitch[4], David Mehringer[5], Raymond Plante[5], Marc W. Pound[1], Kevin P. Rauch[1], Peter J. Teuben[1]

[1] *University of Maryland, College Park, MD*

[2] *California Institute of Technology/ Owen's Valley Radio Observatory*

[3] *University of California, Berkeley, CA*

[4] *University of Chicago, Chicago, IL*

[5] *NCSA/University of Illinois, Urbana-Champagne, IL*

Abstract. The Combined Array for Research in Millimeter-wave Astronomy (CARMA) will be the combination of the BIMA, OVRO, and SZA millimeter arrays, at a new high elevation site (7200', 2200 m). With first light scheduled for 2005, CARMA will be the first heterogeneous millimeter array, combining antennas with diameters of 3.5, 6.1 and 10.4 meters.

CARMA's monitoring system (CAM), is designed to collect monitor data periodically from a set of heterogenous systems and organize these monitor points hierarchically. It allows for the collection/collation of sampled values from arbitrary monitor points and the storage of such monitor point values in files and databases. Monitor points may be mapped to multiple hierarchies, for example, physical hierarchies like antennas, receivers, and drives, or logical hierarchies like sub-arrays and antennas.

The monitor system was designed to accommodate monitor points associated with state machines, physical quantities (such as voltages), complex quantities (such as phases) and logical quantities like process health. It is also designed to help identify and isolate problems quickly, especially during integration at the new high site.

1. Design Considerations

The Combined Array for Research in Millimeter Astronomy (CARMA) is a project undertaken jointly by the California Institute of Technology (Owen's Valley Radio Observatory), the University of California (at Berkeley), the University of Illinois (Urbana-Champagne), the University of Maryland (College Park), and the University of Chicago. CARMA will combine two existing arrays, the Berkeley-Illinois-Maryland array (BIMA) and the array at Owen's

Valley Radio Observatory (OVRO), and a new radio millimeter array, the SZA, being built at the University of Chicago.

The design of the CARMA Monitoring System (CAM) was shaped primarily by the need to facilitate development by teams distributed across multiple institutions, and by anticipated usage of monitor data.

Specifically, the design was governed by the following:

- CARMA is a collaboration between multiple institutions, each developing software and/or hardware at their respective locations.
- The Array Control Computer will require a uniform stream of monitor data from all of the different types of antennas, each with its own hardware and software.
- Minimal effort should be required to integrate CARMA monitoring data to existing methods of displaying health information for operating arrays such as OVRO and BIMA.
- Some monitor data will be used for correcting visibility data.
- Monitor data should provide the observers and engineers who operate and maintain the array with the information required to perform their functions.
- Allow for independent evolution of the various portions of CARMA (antennas, correlators, and other subsystems) by decoupling major subsystems.

2. The CARMA Software System - An Overview

The CARMA Software System conceptual design is based on the principle that the software must reflect the instrument being manipulated (the array) and must also provide the capability to use the instrument for radio astronomy. As Figure-1 shows, the Observer Object encapsulates knowledge about astronomy and the CARMA Array Object(s) encapsulate knowledge about the instrument. The Array Object communicates with the various subsystems like the antennas and the correlators through standardized interfaces using CORBA services. This insulates the Array Object and the array control subsystem from heterogenous hardware and software.

3. The CARMA Monitor System - An Overview

Figure-2 shows the architecture of the CARMA monitor system. It is designed to provide a uniform framework for

1. writing sampled monitor data onto the monitor stream
2. collecting all monitor data from each subsystem into a subsystem frame,
3. sending the subsystems frames to the array control computer every half-second, and
4. presenting CARMA monitor data to processes within the array control computer.

The CARMA Monitor System provides a half-second heartbeat for the array as a whole, and presents monitor data from heterogenous hardware and software, in a structured, uniform way, to processes within the array control computer.

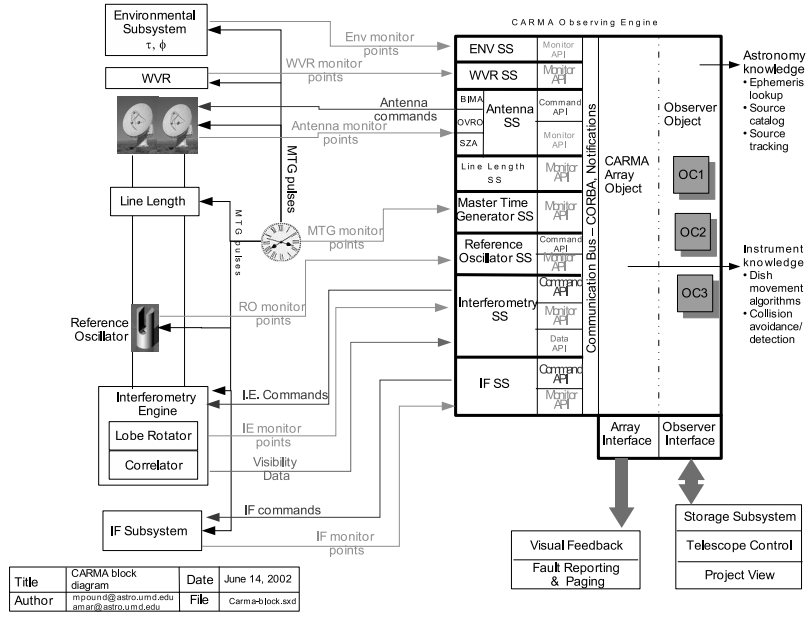

Figure 1. A Block Diagram of the CARMA Software System. The dark arrows represent commands, and the lighter arrows represent monitor and visibility data

4. Monitor Points

The monitoring system is composed of Monitor Points whose sampled values are monitor data. Monitor data is any data produced by the array excepting the visibilities (auto and cross spectra produced by the correlators) and a limited set of data produced by procedures that are run on demand (for example, optical pointing). Monitor data are collected and sent to the Array Control Computer (ACC) every half-second. Monitor data include values produced directly from hardware as well as copies of commanded hardware states. Monitor data values such as total power and calibration device state are used in the processing of correlator data.

5. Monitor Hierarchy

Monitor points are arranged in a hierarchy to provide structure to the large numbers of Monitor Points in the monitor system. the levels in the hierarchy can be purely logical constructs, such as an Antenna Local Oscillator System, or they may represent physical devices (called MonitorDevices) such as a Gunn oscillator. The root of the hierarchy is "CARMA Monitor System" which represents the collection of all monitor points across the entire CARMA system. An example of programmatic use of the hierarchy and API is:

```
CarmaMonitorSystem carmaMon;
Gunn3mmMon gunn = carmaMon.OVROantenna(4).LOsystem().Gunn3mm();
```

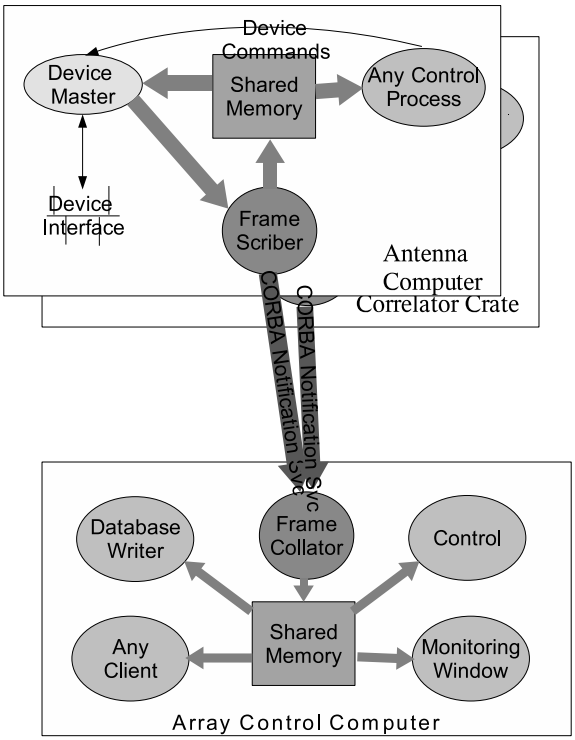

Figure 2. A Block Diagram of the CARMA Monitor System.

```
float tunerPosition = gunn.TunerPosition().getValue();
```

6. Monitor Data Flow

A monitor subsystem is any collection of CARMA components that communicates with the ACC through a single computer (a Linux box). Within each monitor subsystem, monitor data are collected into a "subsystem monitor frame". Each such frame is sent to the ACC using a CORBA Notification channel. A collator process in the ACC waits for such notifications and collates the frames received from all subsystems into a consolidated CARMA system monitor frame. This CARMA system frame is then made available to every process on the ACC. This process of sending notifications and collation/creation of a system frame occurs every half-second.

References

Gwon, C. 2004, this volume, 708
Scott, S. L. 2004, this volume, 768