

## The CARMA Control System

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

<sup>1</sup> *University of Maryland*

<sup>2</sup> *California Institute of Technology/Owens Valley Radio Observatory*

<sup>3</sup> *University of California, Berkeley*

<sup>4</sup> *University of Chicago*

<sup>5</sup> *NCSA/University of Illinois*

**Abstract.** The Combined Array for Research in Millimeter-wave Astronomy (CARMA) will be the combination of the BIMA, OVRO, and SZA millimeter arrays. With first light scheduled for 2005, CARMA will be the first heterogeneous millimeter array, combining antennas varying from 3.5 m to 10.4 m in diameter. The controls for CARMA involve creating a uniform interface for all antennas. The antennas are grouped into five independently-controlled sub-arrays, which will be used for scientific observations, engineering, or maintenance. The sub-arrays are controlled by two components: the Sub-array Command Processor (SCP) and the Sub-array Tracker (SAT). While each sub-array has a dedicated SCP for handling command processing, a single SAT computes and distributes slowly varying parameters to the necessary sub-arrays. The sub-array interface uses CORBA distributed objects to physically separate the user interface from the array. This allows for stability in the core engine controlling the array while enabling flexibility in the user interface implementation.

### 1. Design Considerations

Four major factors were considered when designing the CARMA Control System. First, being a multi-institutional collaboration, each developing hardware and software at their respective locations, it was necessary to have a design that would allow institutions to develop independently and in parallel. Second, the array will consist of eight 3.5 m, nine 6.1 m, and six 10.4 m antennas; to deal with the heterogeneous nature of the array, the antennas will be treated as distributed objects, each having a uniform interface for the system. Third, having a heterogeneous array means that data collection will differ for different antennas; the system must therefore be able to take such differences into consideration in

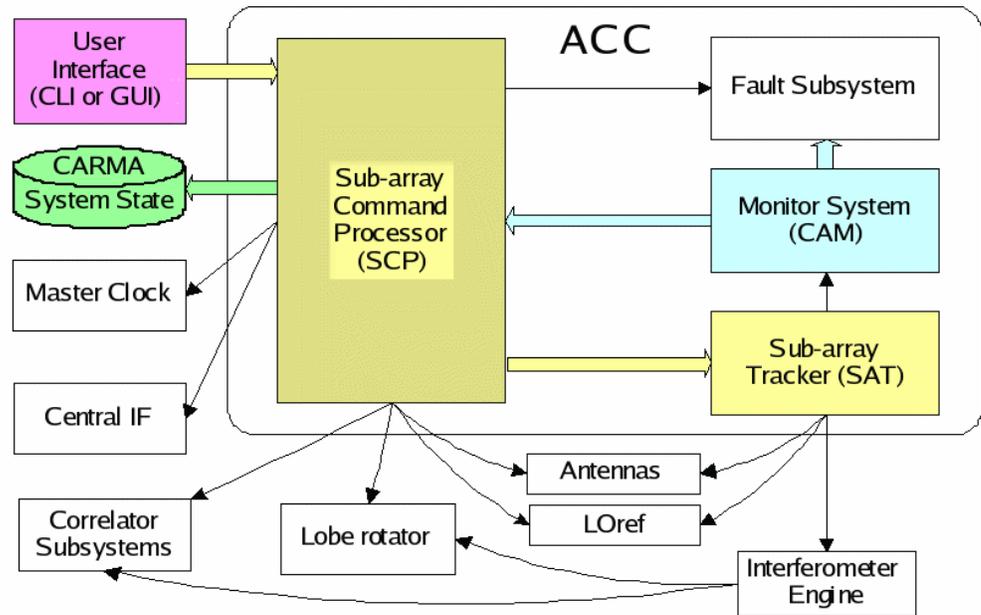


Figure 1. Control System Interactions: Control system components are the SCP and the SAT. Commands flow from a command line input (CLI) or a graphical user interface (GUI) to the SCP, and from the SCP to the SAT. Interactions between control system components and other subsystems are shown with black arrows.

various circumstances. The last factor was to have a system that would allow remote access to the array controls.

## 2. Sub-arrays

A sub-array is a logical grouping of antennas based on scientific, engineering, and maintenance purposes. There are five sub-arrays for the control system; each can be interactively controlled and are distinguished by a correlator and a reference oscillator. The five sub-arrays are

- Science #1: 15 station spectral correlator for the 6.1 m and the 10.4 m antennas, reference local oscillator #1
- Science #2: 8 station wideband correlator for the 3.5 m antennas, reference local oscillator #2
- Engineering #1: Reference local oscillator #3, used for solving frequency-related problems
- Engineering #2: Passive reference local oscillator, used for solving problems not related to frequency
- Offline: For antennas that are offline

With the exception of antenna assignment and shadowing calculations, the schedule and command processing for each array is independent.

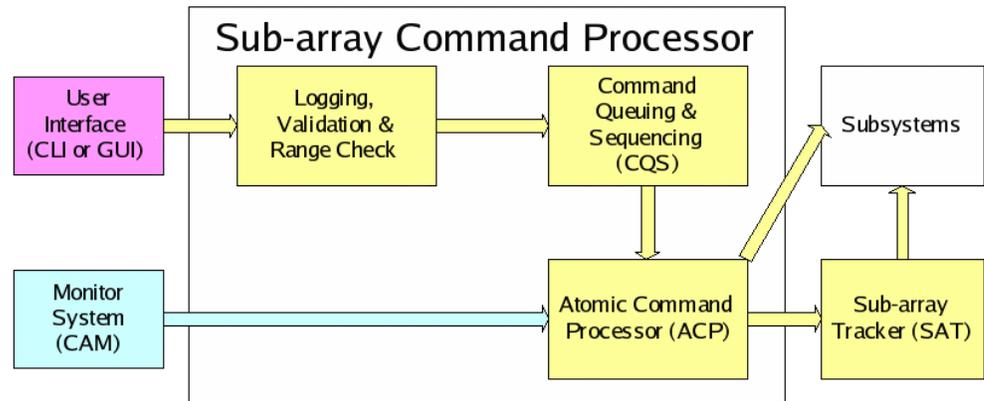


Figure 2. Sub-array Command Processor (SCP): Command input from a CLI/GUI are logged and checked for valid ranges and syntax. It is then queued for passing to the ACP. The ACP then passes commands to the SAT or to other subsystems.

Commands for a sub-array contain all methods necessary for controlling antennas, the interferometric array, and the correlator. They consist of three basic types: setups, queries, and procedures. Setup commands change the state of the sub-array, typically completing within 200 ms. Queries return values of specific monitor points, or sets of monitor points. Procedures involve time consuming operations, such as data collection. All commands pass through a queuing and sequencing mechanism, which enable the array to run schedules automatically while also allowing interactive use. Commands are also used to define the CARMA System State (CSS), which is necessary to restore the array to its previous state in case of system restart.

### 3. Implementation

Distributed objects (DO) are physically throughout the array. Because of this, CORBA is used to communicate between the DOs. The sub-array interface itself represents a CORBA DO, whose methods are called to execute control commands. The two control components of the sub-array are the Sub-array Command Processor (SCP) and the Sub-array Tracker (SAT). The SCPs and SAT all reside in an Array Control Computer (ACC); the centralization of these components on a single system results from synchronization issues and maintenance considerations.

#### 3.1. Sub-array Command Processor (SCP)

The SCP is the major control component, through which all control input passes via the methods of its DO, the “Control API” (CAPI). Each sub-array will have a dedicated SCP for handling all control input commands. Commands from a user interface are checked for valid value ranges and then queued before being

processed by the Atomic Command Processor (ACP) within the SCP. The ACP does the actual execution of the command, which involves calling commands on the various DOs in the system. All methods of the CAPI implement a command and response protocol by sending a return value, which contrasts the rest of CARMA, where there are no return values. Return values are sent when a command is queued or executed.

### **3.2. Sub-array Tracker (SAT)**

The SAT ensures that the distributed components receive updated information to correctly track sources. There is only one SAT for all the SCPs; the SAT receives input from the SCPs and periodically recalculates the relevant system parameters, sending them to the appropriate sub-arrays. For example, the local oscillator reference receives updates in the local oscillator frequency based on current source velocity; antennas receive right ascension (RA) and declination (DEC) measurements, which are converted to local azimuth and elevation.

### **References**

- Amarnath et al. 2004, this volume, 720  
Plante et al. 2003, in ASP Conf. Ser., Vol. 295, ADASS XII, ed. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook (San Francisco: ASP), 269  
Scott et al. 2004, this volume, 768