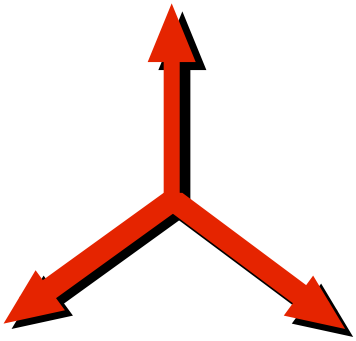
EXPANDED VERY LARGE ARRAY



System Block Diagram

A24051Nxxxxx Rev 1.0

January 9, 2004

EVLA Software Design Group:

R. Moeser, K. Ryan, K. Sowinski, B. Waters

NATIONAL RADIO ASTRONOMY OBSERVATORY
P.O. Box 0, Socorro, New Mexico 87801

Operated by Associated Universities, Inc.
Under Contract with the National Science Foundation

Revision History

Date	Version	Description	Author
09 Jan 2004	1.0	Initial Release	Edited by K. Ryan

Table of Contents

1 Introduction

A team of software engineers has been assembled to create an overall design of the EVLA system software from proposal to archive.

1.1 Purpose of this Document

The purpose of this document is to convey our idea of the system at the highest level. Changes after the review meeting on Jan 5, 2004 have been incorporated.

The first goal was to discover the major functional components of the system. Since this first step represents the foundation upon which the rest of the system will be based, it is important that reviewers agree to the correctness of this analysis before proceeding with further design.

1.2 Methodology

The system was divided into three functional areas:

1. Observation Preparation
2. Observation Execution
3. Observation Analysis

The areas were populated with components gleaned from current diagrams and requirements specifications.

Lines were drawn between components to represent information flow. Interfaces are not resolved further; only very broad descriptions of data flow between components are given.

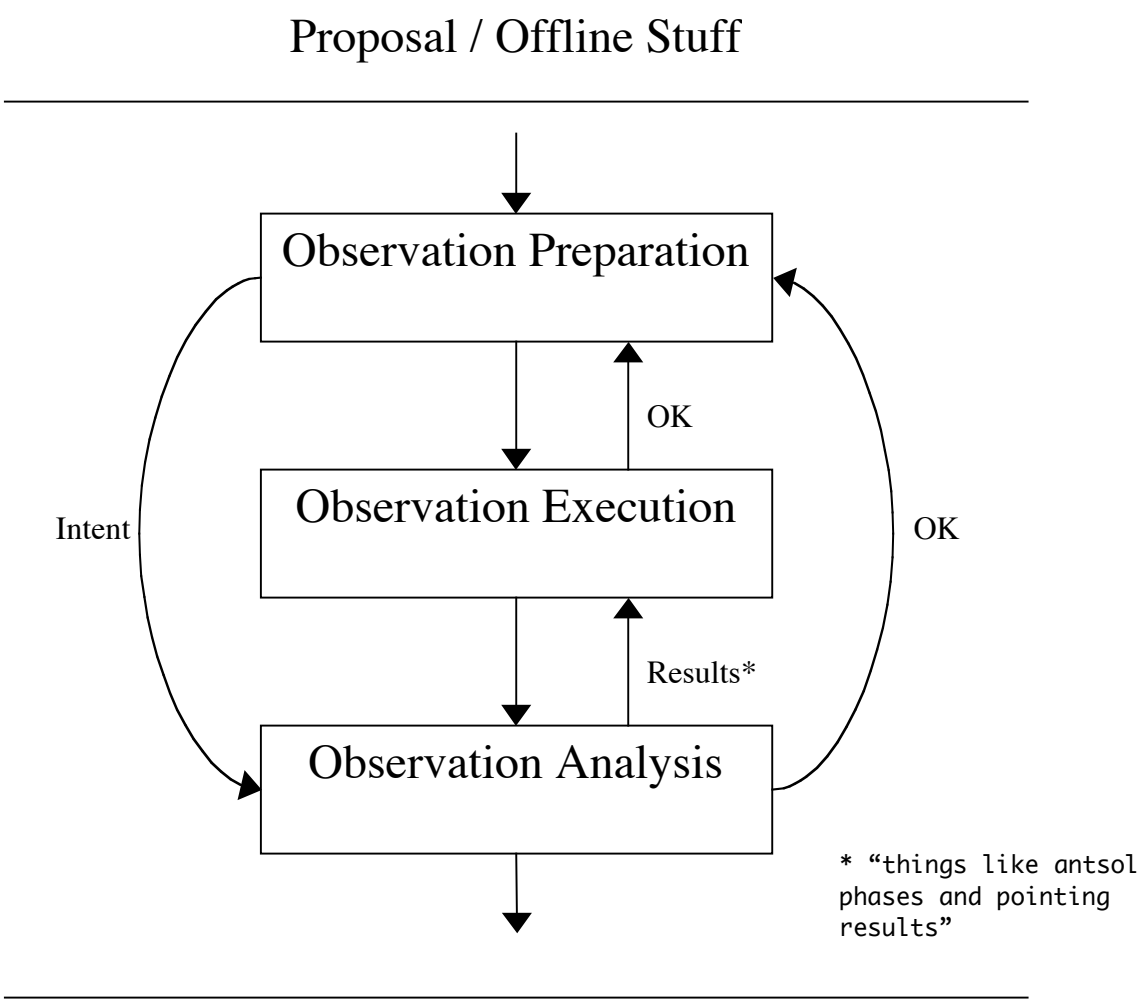
2 Overview Diagram

The purpose of this diagram is to allow us to separate our domain from others, and to describe the function of the system at the highest reasonable level.

At this level, the system we are contemplating will accept information from the astronomer and/or the proposal tool and provide data to the Archive for post processing both in the short term for immediate feedback to the system and to users, and in the longer term for astronomical purposes.

On the diagram lines convey needed information sources and sinks, but do not prescribe any particular communication method or path.

Observation Preparation is supposed to include transforming a Program block into a collection of Scheduling Blocks, and choosing from all submitted Program blocks the "best" Scheduling Block (or blocks) to be run at any moment. 'Intent', as we have chosen to call it, must be made available to the Observation Analysis function if it is to have any hope of understanding the dataflow from the correlator.



Post Processing

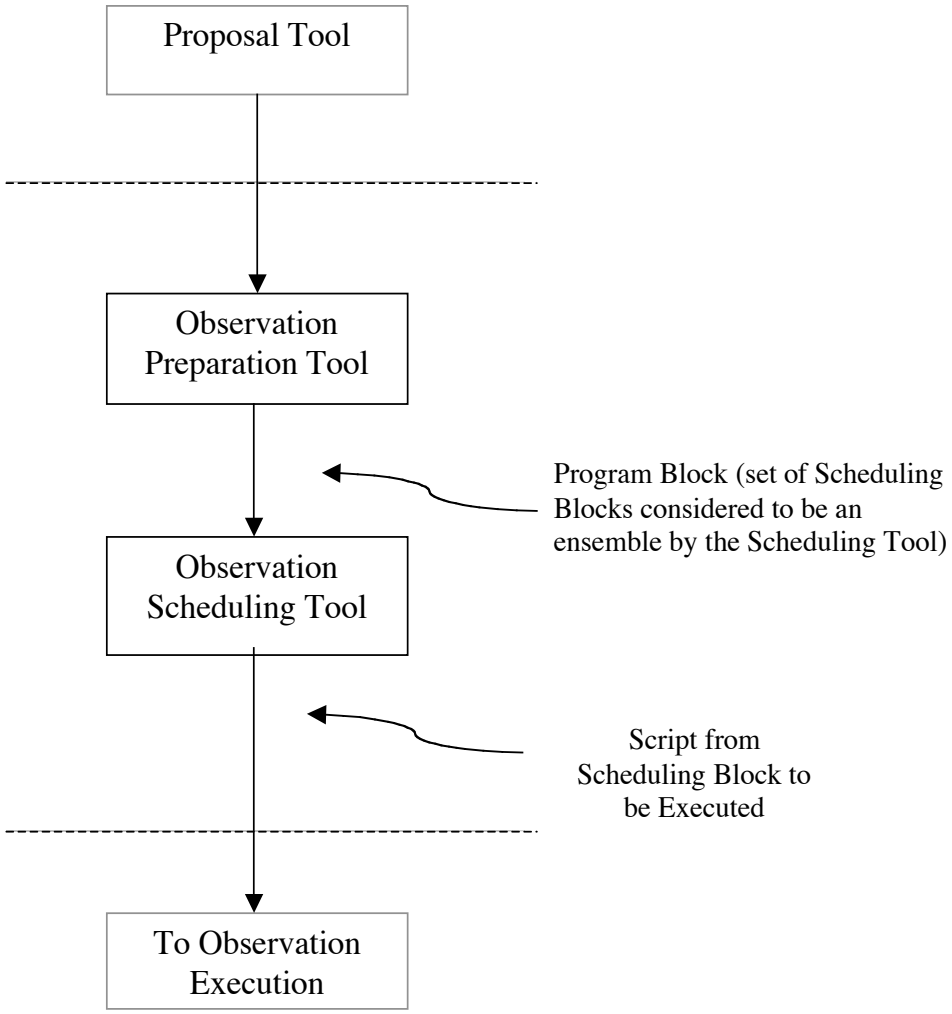
The Observation Execution function accepts scripts that contain sufficient information to control the array to achieve some astronomical goal. The goal is opaque to this level of the system. It only knows enough to organize the correlated data and pass it, and all ancillary and metadata, along to the Observation Analysis layer. The fact that a script successfully terminated should be available to the scheduler.

The Observation Analysis subsystem is where it all comes together. Correlated data is accepted from the correlator backend processor, and necessary descriptive data is provided from the monitor and control system. Three important functions occur here: first, the data is placed in an Archive. Secondly, derived information needed by the observation in progress is returned to the monitor and control system. Finally, astronomers and operators will have access to data, processed in meaningful ways, to enable judgments about the integrity of the observation or the instrument.

The workings of the system will be made visible at all levels through displays for the operator or other users.

3 Observation Preparation Diagram

The Observation Preparation system is further resolved into two sub-components - the Observation Preparation Tool and the Observation Scheduling Tool.



3.1 Observation Preparation Tool

The Observation Preparation Tool takes input from the user and from the Proposal Tool, and generates a set of Scheduling Blocks that it can submit to the Observation Scheduling Tool.

The Observation Preparation Tool is (perhaps) the only sub-system in this presentation that does not necessarily have any hard-real-time requirements; it is not a part of the "online" system.

However, we intend to describe the data model for observation descriptions which are managed by this tool, and to fully define the Scheduling Block and the Control ScripT. That is, we intend the scope of our analysis to include the full definition of the inputs and outputs of the Observation Preparation Tool, and so it appears on this diagram, rather than being hoisted up to the less-well-resolved "offline" layer above.

3.2 Observation Scheduling Tool

This second part of the obs-prep system is (mostly) the Dynamic Scheduler. (We say "mostly" because there are things that provide input to the scheduler, such as site instrumentation, which are not resolved in this diagram.)

The Observation Scheduling Tool takes as input a set of Scheduling Blocks, picks the best one, and submits the Control Script inside the selected block to the Observation Execution layer.

4 Observation Execution

The Observation Execution sub-system performs the actual tasks of moving the antennas of the array, configuring the electronics, and driving the correlator to produce (among other things) visibilities.

As such, the Observation Execution environment utilizes the Monitor and Control (M&C) systems for the Antenna (AMCS), Correlator (CMCS) and the Correlator Back-End (CBE).

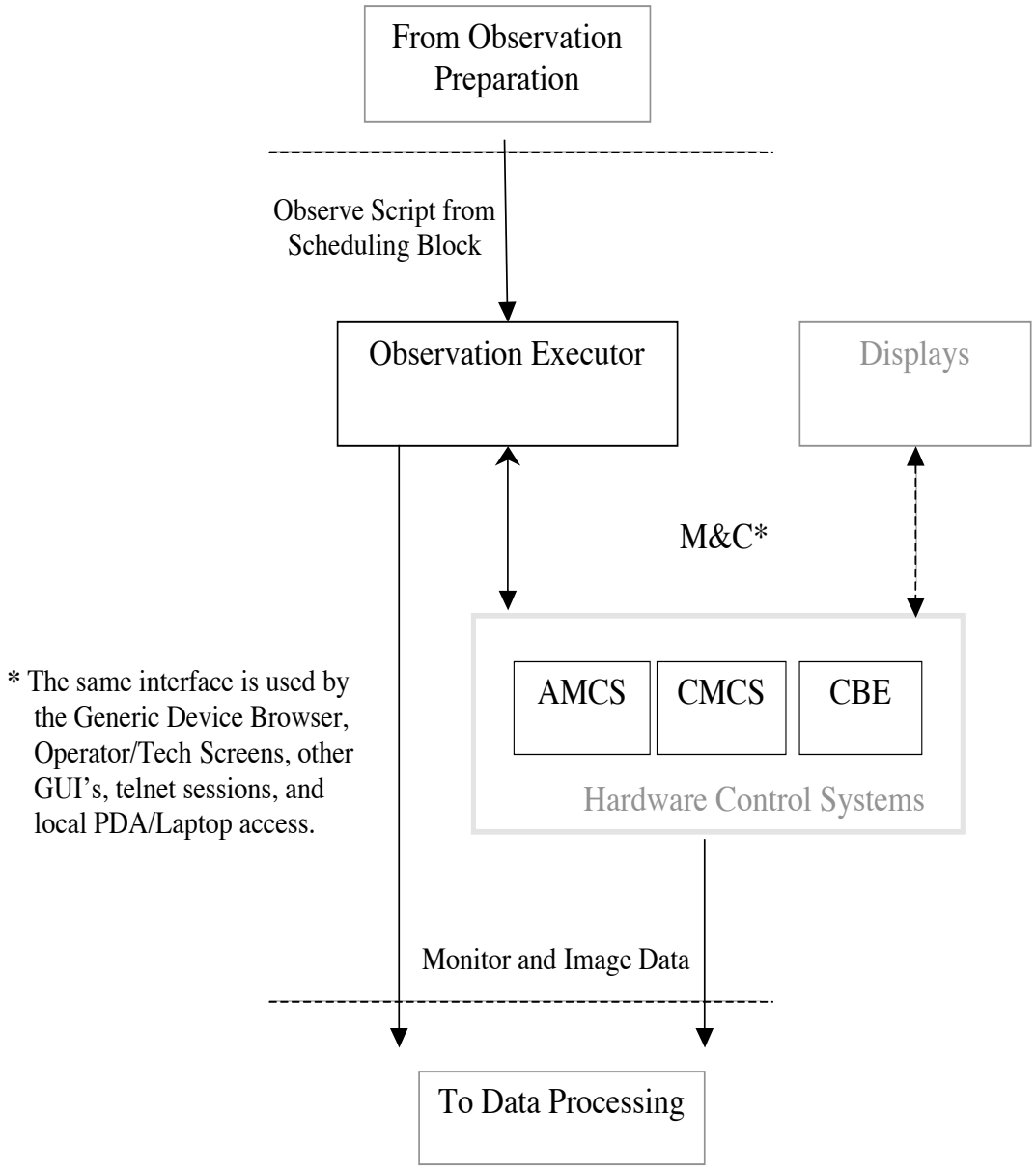
For the purposes of this discussion, we focus on that part of the Observation Execution system that accepts Observe Scripts from the Observation Scheduling Tool, and feeds results into the Observation Analysis layer.

4.1 Observation Executor

The Observation Executor is an interface layer that mediates between the Observation Scheduling system and the hardware control systems:

The Executor translates requests expressed in Observe Scripts into hardware configuration commands that are understood by the Hardware Control Systems. (That's the line from the Executor to the hardware control systems -- a command interface from the Executor to the M&C, and a monitor/feedback interface going the other direction.)

The line from the Executor to the Observation Analysis layer represents configuration information and other data required by Observation Analysis.



4.2 Other Monitor & Control Clients

Other sub-systems, such as the operator Displays, may communicate with the M&C system; they do so through the same M&C interface that the Executor uses.

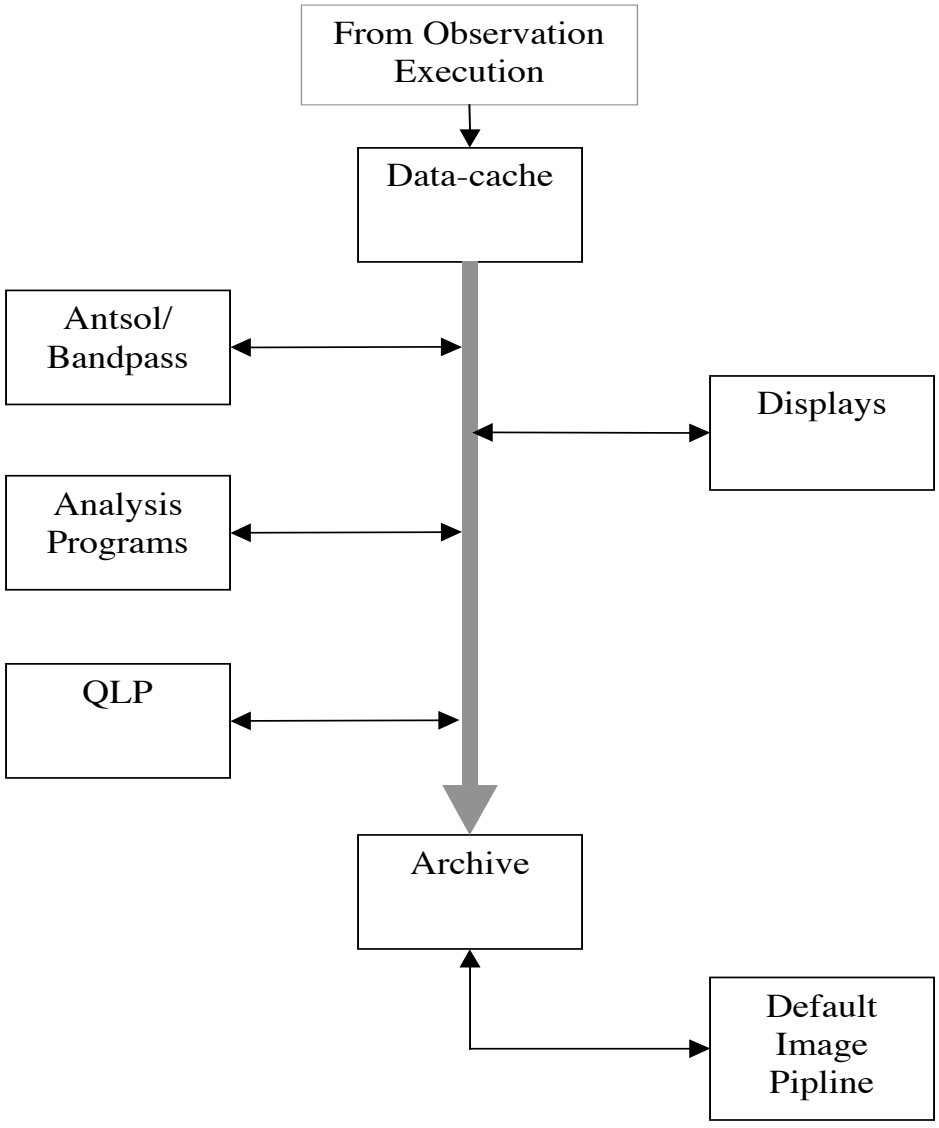
4.3 Hardware Control Systems

The commands come in from the Executor or (for example) an Operator shell session; the resulting visibility data flows through the Correlator and the CBE and on down to the

Observation Analysis layer. And of course, the M&C system provides monitor data to the Observation Analysis layer as well.

5 Observation Analysis

There is some method in the placement of boxes. Process flow is generally from top to bottom. Storage/Data is in the middle; things happen on the left; everything can be examined on the right. The bottom right box is left there for the moment as a semi-orphan.



We have distinguished a Data Cache and an Archive without saying how they are to be implemented. They may in fact be one and the same physically, but the boxes reflect different intended use. The Data Cache accepts data from the CBE and all such that the

monitor and control system cares to provide. The Data Cache will be resident at least as long as a Scheduling Block is active; perhaps as long as a group of related scheduling blocks are active. It is intended to be of modest size and inherently fast so that all of the processes on the left are not limited by access latencies. The persistent Archive is just that: everything will eventually be in it and it lasts, in multiple, distant copies, forever. The Default Image Pipeline will deal only with the persistent Archive.

Displays subsumes both the "What's up" system of the SSR and operator displays. We think it likely that one thing can serve both audiences. The display system will have access to data in any of the stores and from results of any of the processes on the left of this diagram. It is the peephole through which everyone will know what is happening with the system.