Expanded Very Large Array	EVLA-SW-??? Revision:1.0 2004-Feb-25 *Presentation Notes* B.Waters

EVLA Software High-Level Design Presentation Notes

EVLA Software Design Group:
T. Morgan, K. Ryan, K. Sowinski, B. Waters

Keywords: EVLA, e2e, Software	
Author Signature: B.Waters	Date: 2004-Feb-25
Approved by: R. Perley, P. Napier Institute: NRAO	Signature: Date:
Released by: Institute:	Signature: Date:

Change Record

Revision	Date	Author	Section/ Page affected	Remarks
1.0	2004-Feb-25	B. Waters, K. Ryan	All	Initial Version

Contents

1 Introduction

1.1 Purpose

This document presents the high-level design of EVLA Computing, with a focus on identifying the major interactions between subsystems that comprise the primary data-flow of the EVLA system. This places each sub-system of EVLA computing in context with relation to the others. The previous design had shown subsystem boxes with almost no lines between them. This time we drew the lines.

At this stage, the data structures which pass between the various sub-systems is yet to be completely resolved, but it is intended that the interactions by which all required data structures are exchanged be accounted for.

If a required interaction does not fit onto the diagrams presented here, please speak up!

1.2 Methods and Milestones

On 5 January 2004 the Design Team presented a rough set of high-level subsystems that encompassed EVLA computing; the interactions between these subsystems was defined at a very high level to outline the primary data-flow from Proposal Submission, through scheduling and execution by the telescope, to the delivery of science data products via a data-reduction pipeline.

The Design Team proposed a set of three iterations of design; each cycle of iteration would follow the data-flow through the system with increasing detail. The first iteration – this one – would serve to identify each subsystem's functionality, and the relationship between the subsystems, by identifying the major interactions that are needed to support the primary data-flow of the EVLA system.

In the second iteration, the Design Team will identify the data that are passed along those lines: the data structures, and the approximate rate (e.g. speed, periodicity) at which these structures are passed between the subsystems.

The third iteration will identify the standard mechanisms by which the data structures are passed: the data-transmission protocols, and how data is transformed from a subsystem context to the data-transmission format (marshalling and serialization).

1. February 26 - First Iteration - Subsystem Context
2. March 18 - Second Iteration - Data Flows Defined
3. April 8 - Third Iteration - Interfaces Defined

These three passes will be compiled into an overall EVLA System Design. We expect the following more substantial reviews with outside participation. Dates are approximate.

1. May 1st 2004 - system design specification + review
2. June 1st 2004 - Conceptual Design review

1.3 EVLA in Context

During this design cycle, the EVLA Design Team was able to better define the relationship between EVLA computing and the other NRAO computing development projects, particularly ALMA and that of the Interferometry Software Division (ISD). At the same time, the NRAO End-to-End (E2E) design effort produced a preliminary analysis of the overall data flow for NRAO computing.

The interface between EVLA and NRAO-E2E computing systems had been an area of concern for the EVLA project. In this cycle, working discussions with ALMA, ISD, and E2E has resulted in a design for data processing

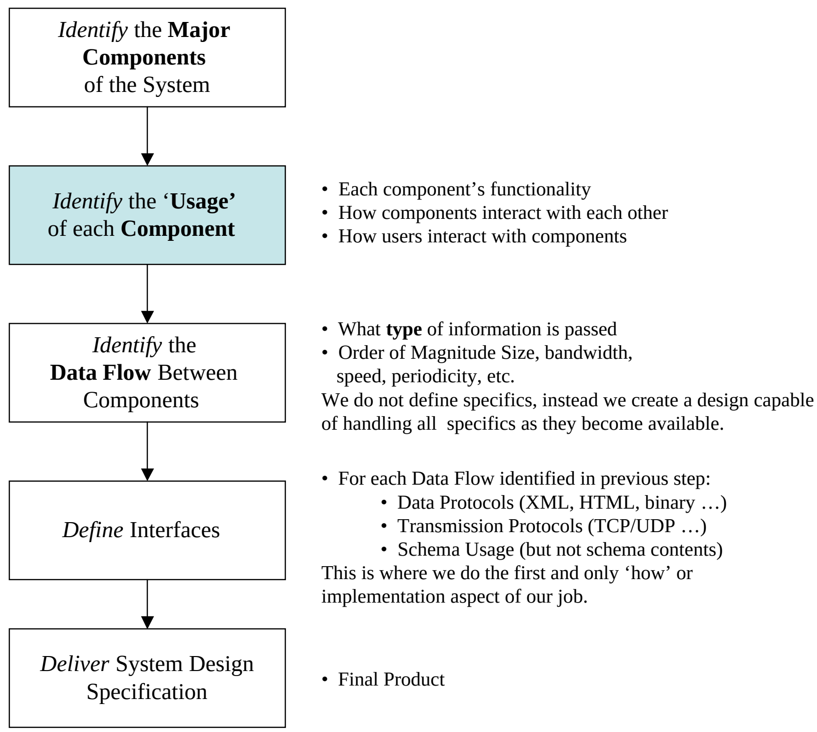


Figure 1: EVLA Design Team Milestones

that has a plausible amount of overlap between ALMA and EVLA, and has better defined the relationship between EVLA and NRAO-wide, E2E data-processing. Although much work remains to be done, a framework for collaboration between EVLA, ALMA, and ISD, in the context of an overall data-flow identified by NRAO E2E, has been set. This is a significant achievment.

This E2E effort will continue in parallel throughout the EVLA design cycle as currently planned (through June 2004).

EVLA is designed to use data-reduction software developed by the Interferometry Software Division (ISD). The ISD is currently developing tools for ALMA, and it has been noted that common terminology for certain terms will be needed between EVLA and ALMA in order to avoid confusion in the further definition of the shared data structures. We have yet to formally agree upon these common definitions, but a brief examination of UML diagrams derived from ALMA and EVLA suggests that such commonality will be forthcoming in the next design cycle.

In Appendix A we define the main entities, used to manage the whole observing process, that are referred to in this report.

2 Overview

2.1 End-to-End Data Flow

EVLA data flow fits into the overall NRAO End-to-End (e2e) data flow as shown in Figure 2. [1]

In this design iteration, we were able to identify components that are held in common with other observatory computing efforts; these have been labeled "NRAO e2e" on the Top-Level Data Flow Diagram in Figure 3. NRAO e2e is the source of a high-level observation description, in the form of a **program**, and is the recipient of the data product resulting from an observation, the **Science Data Model**.

The EVLA Top-Level Data Flow Diagram depicts this *primary* data path, from **Program**, through scheduling and execution, to **Science Data Model**, as arrows leading from the top of the diagram to the bottom, down the center of the page.

Other required data flows are shown at the sides of the subsystem boxes; in general, the left side of the box is input, and the right side of the box is output, although this is not always the case. Most of these (secondary?) flows serve to provide feedback between subsystems that influence the execution of the primary data flow.

2.2 Transition Issues

We have begun to think about how to fit the future EVLA system into the near term future when there are still VLA antennas and the WIDAR correlator does not yet exist. There are three distinct periods:

1. before the new correlator
2. a comissioning period, with both correlators are in use
3. when the old correlator is retired.

As far as our design has progressed this is reflected only in the realtime heart of the system. The AMCS will understand both old and new antennas, and the CMCS will be able to understand both the old and the new WIDAR correlator. The output of the old correlator system will be piped to throwaway software which we have descriptively dubbed 'old RTCAT' for obvious reasons. This RTCAT will be provided by the EVLA project and is necessary to enable operation of the VLA in the interim period until the WIDAR correlator and the corresponding RTCAT is ready.

[1] This preliminary, draft e2e diagram was stolen from Doug Tody.

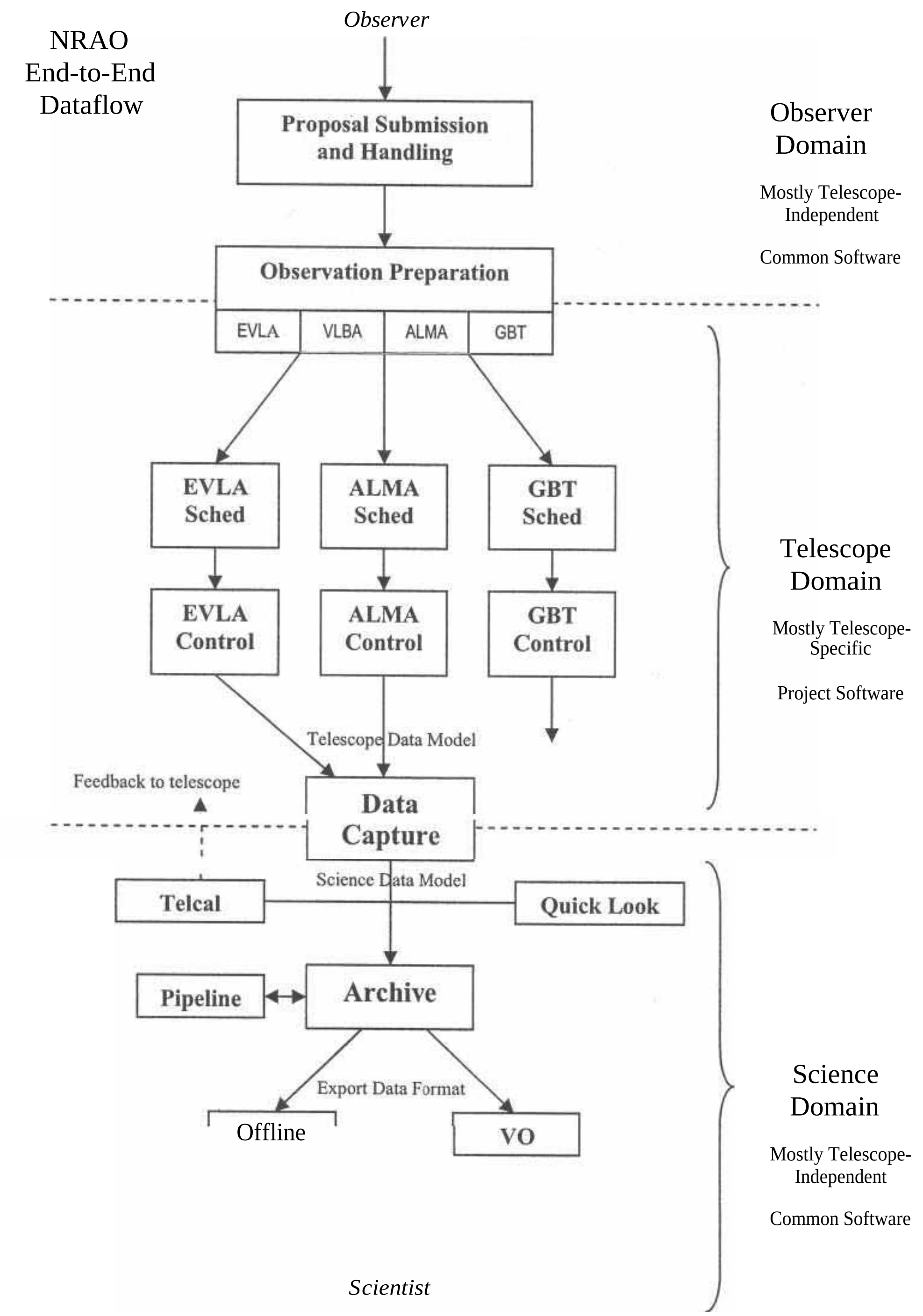


Figure 2: **Draft of the NRAO End-to-End Data Flow Diagram** *from Doug Tody*

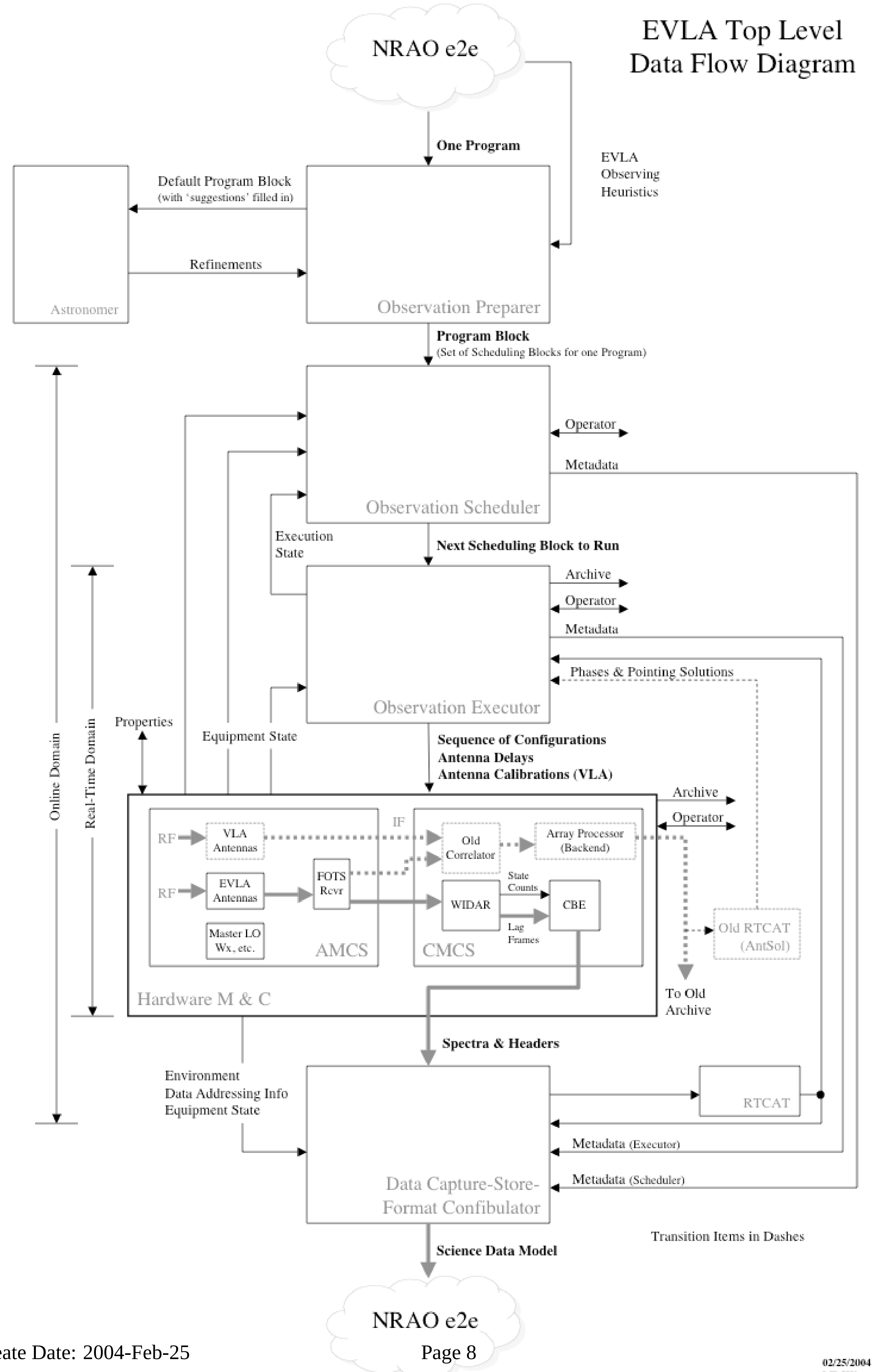
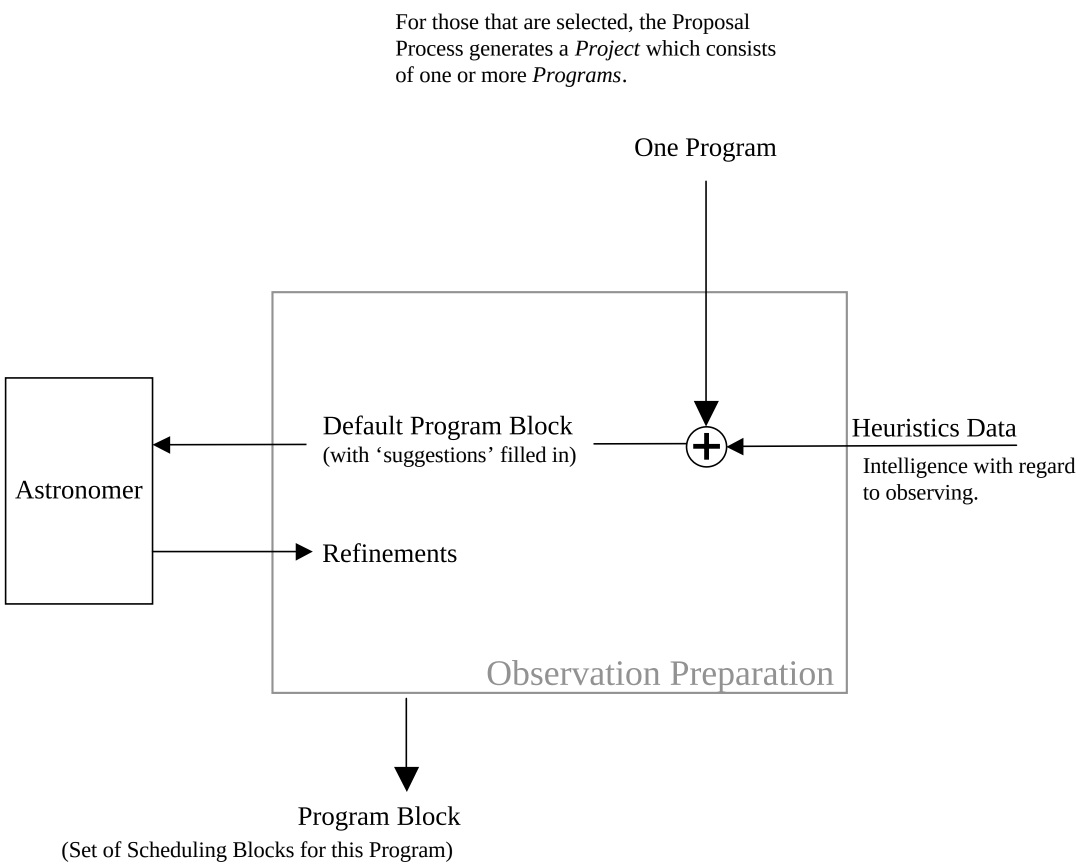
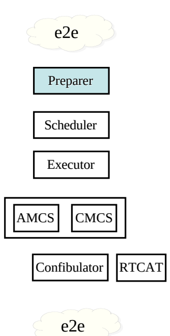


Figure 2: **EVLA Top Level Data Flow Di**

3 Detail Diagrams

In the diagrams which follow, each of the major-subsystem boxes is shown individually, permitting us to focus on the inputs and outputs of each subsystem.





Observation Preparation Tool

What it does:
- Creates a Program Block for a Program

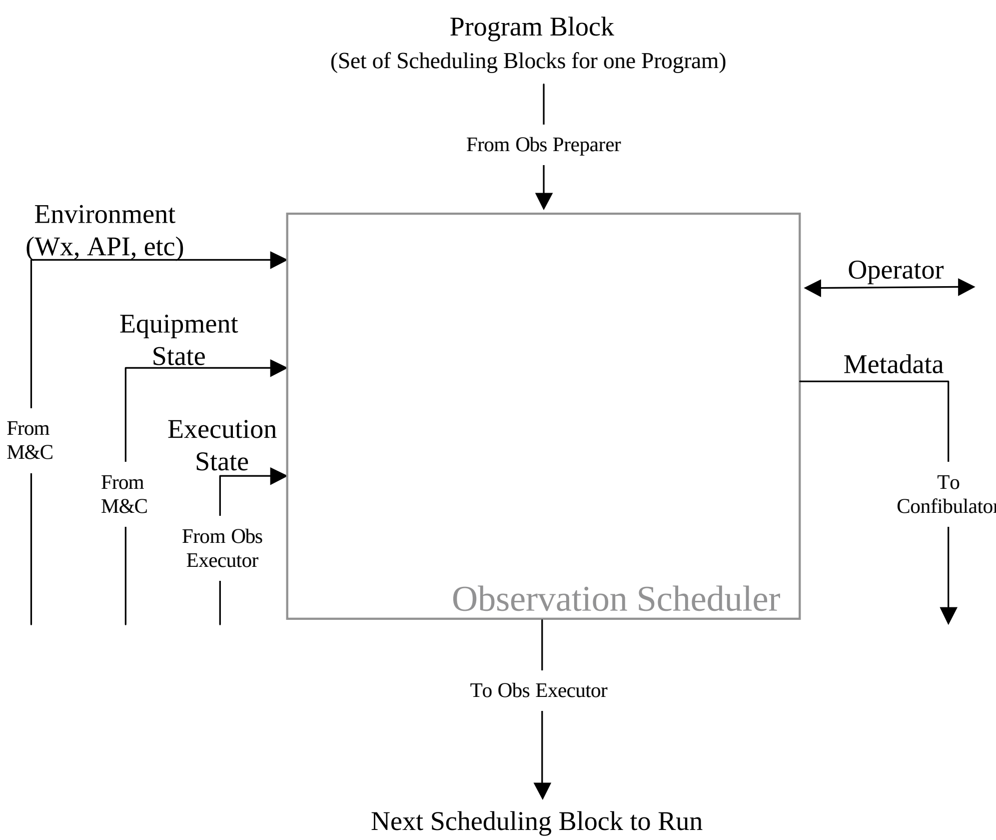
Inputs:
- Program Data (from Proposal Process)
- Observation Heuristics Data
- Astronomer Interaction (Editing/Validation Session)

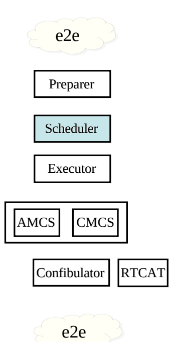
Processing:
- Get Relevant *Program*
- Apply Heuristics
- Present Suggested *Program Block*
- Interact with Astronomer to fine-tune *Program Block*

Outputs:
- Program Block

02/25/2004

Figure 4: Observation Preparer





Observation Scheduler

What it does:

- Dynamically Selects the next Scheduling Block to Execute

Inputs:

- Program Block from Observation Preparation
- Operator - fill queue, override selection …
- Environment (weather, …, …?)
- Execution State (from ?)

Processing:

- Get program blocks under consideration
- Apply environment variables
- Pick next scheduling block to be executed

Outputs:

- One Scheduling Block (to Observation Executor)

02/25/2004

Figure 5: Observation Scheduler

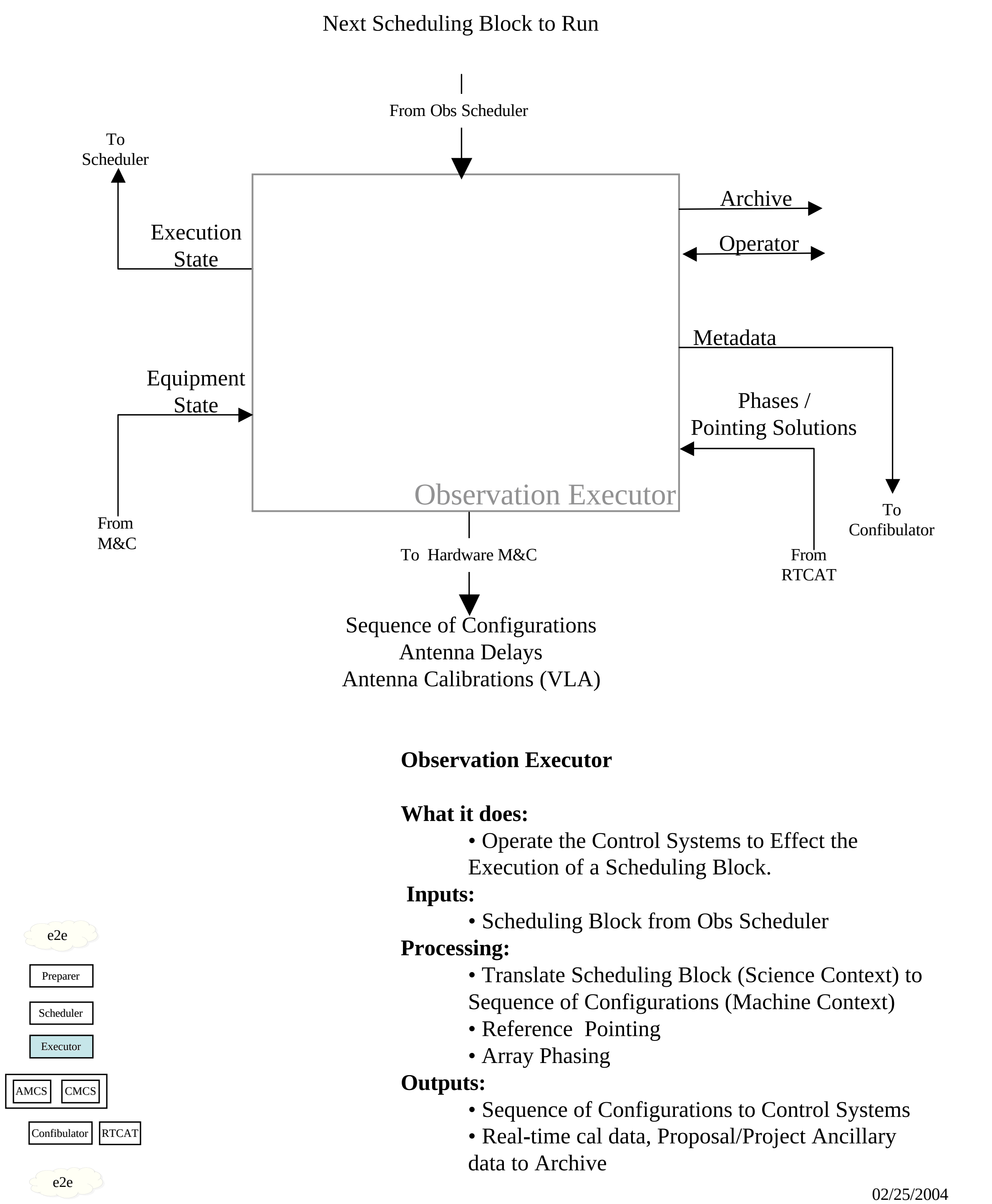
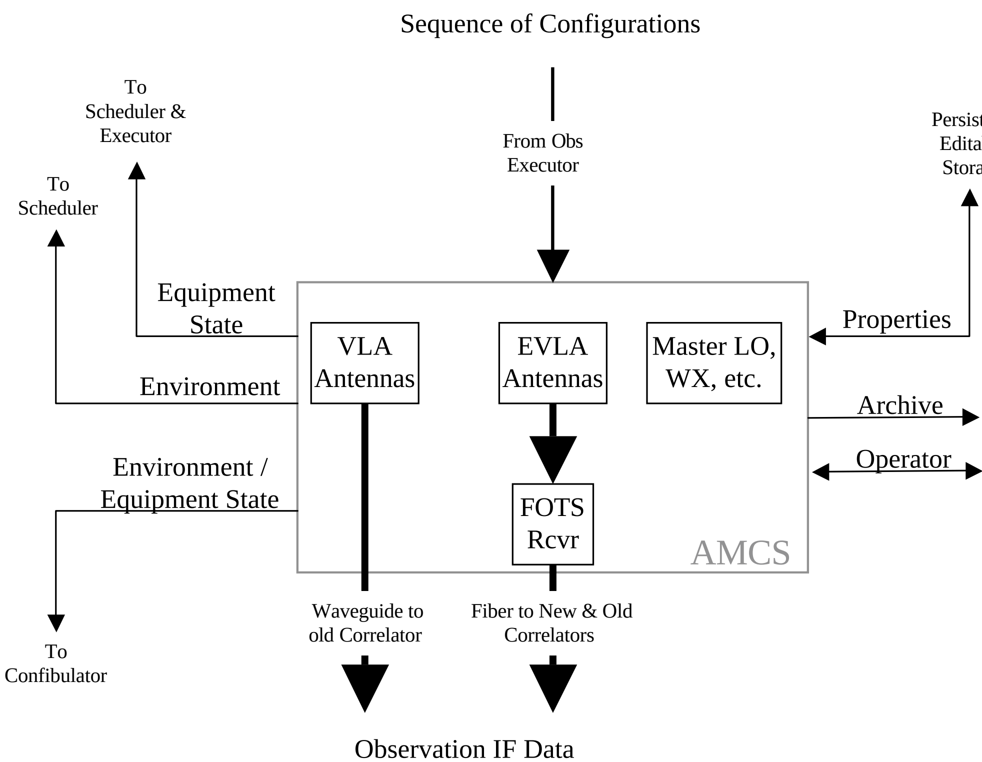
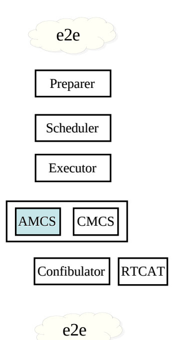


Figure 6: Observation Executor





Antenna Monitor & Control Subsystem

What it does:
- Control System for Antennas and Ancillary Equipment (Weather Station, Master LO, etc.)

Inputs:
- Properties and Parameters (Device, CP, MP, etc)
- Sequence of Configurations
- Individual ‘Commands’
- RF (Observation)

Processing:
- Control Associated Hardware

Outputs:
- Archive Monitor Data
- Observation IF Data

02/25/2004

Figure 7: Array Monitor and Control System

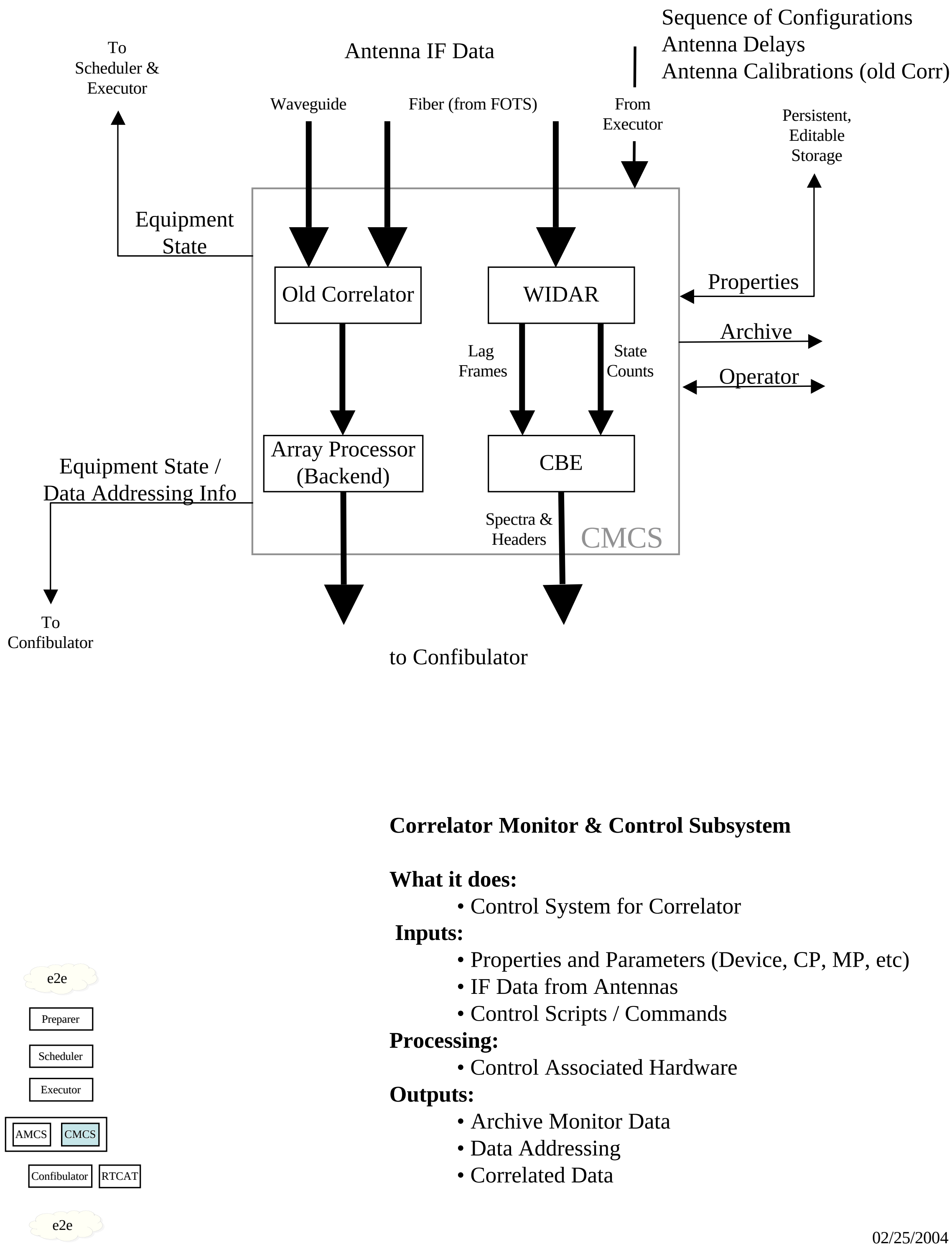
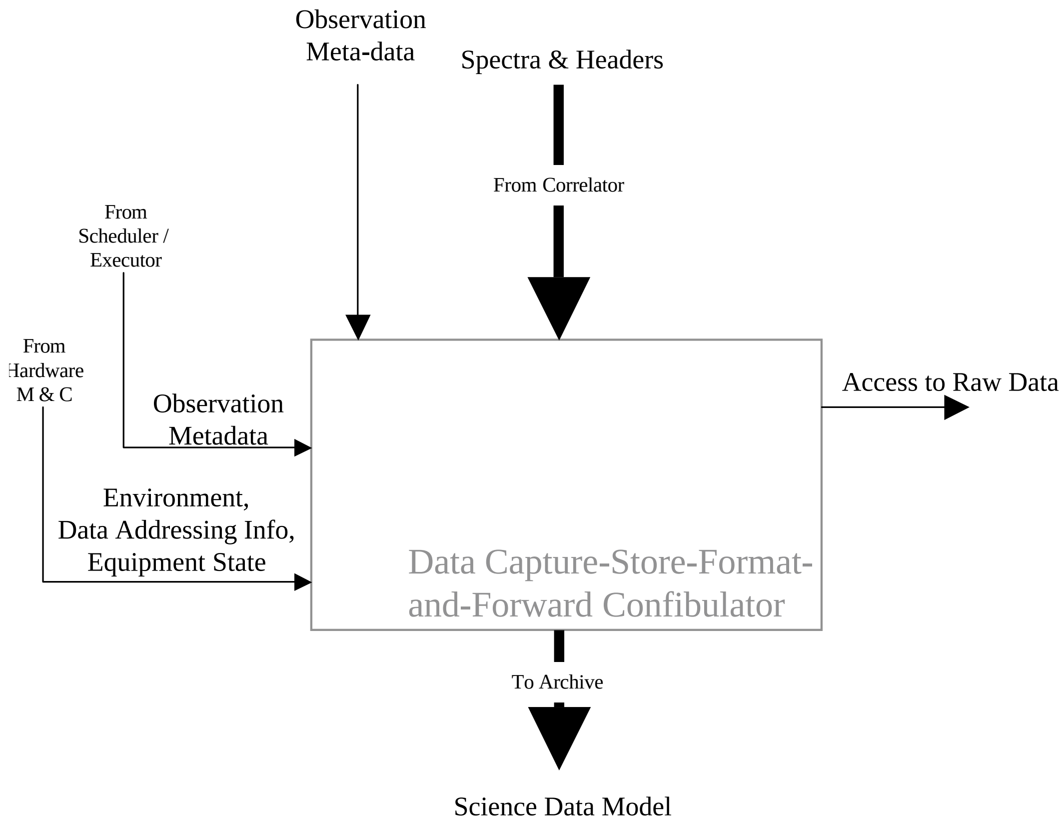


Figure 8: Correlator Monitor and Control System



Confibulator

What it does:

- Buffers Real-Time Data
- Builds Science Data Model

Inputs:

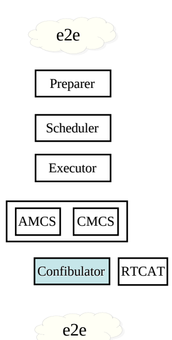
- Correlated Data
- Observation Meta-data

Processing:

- Collation of Data

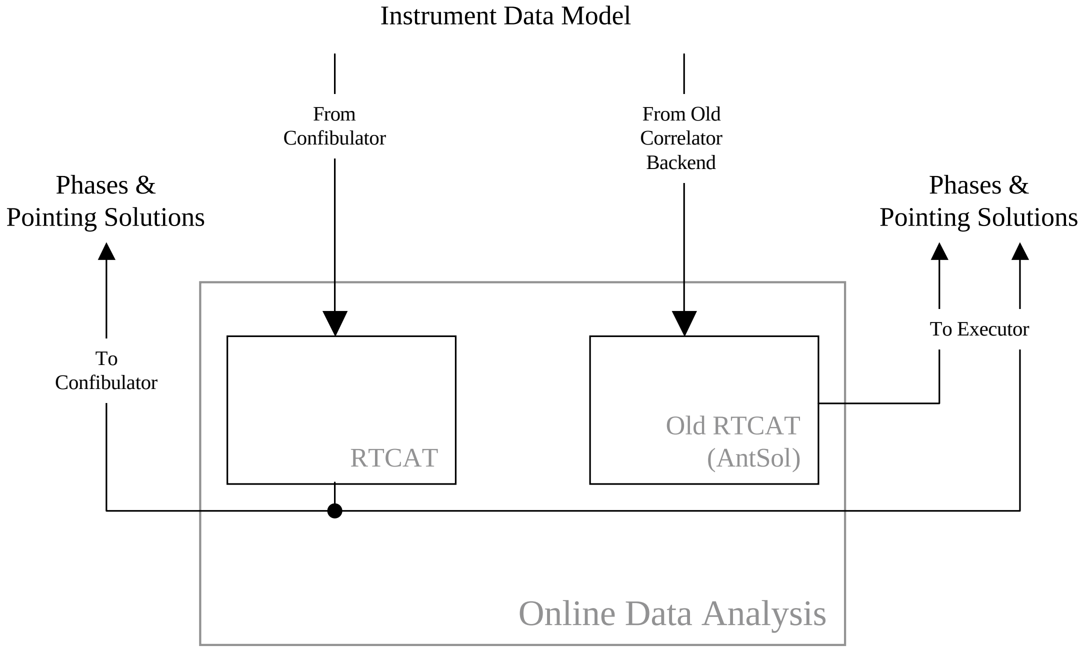
Outputs:

- Science Data Model



02/25/2004

Figure 9: Data Capture



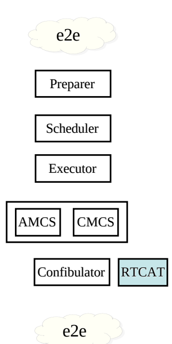
Online Data Analysis

What it does:
- Provide info for an array, in real-time, based on on-going calibration observations.

Inputs:
- Correlated data,
- Metadata.

Processing:
- AntSol & analysis of pointing scans.

Outputs:
- Phases and Pointing Solutions.



02/25/2004

Figure 10: On-Line Data Analysis

A Definitions and descriptions

The SSR introduced its glossary with these words. “In this Section we introduce the main entities, used to manage the whole observing process, that are constantly referred to in the requirements. These entities have some kind of hierarchical structure that will be further refined at the analysis stage of software development.” This document, which is clearly based on SSR terminology, is the first step of the refinement process, an EVLA software glossary of terms which are important at the highest level. The purpose of this list is to provide the design group wih a vocabulary to enable consistent discussions. The present version is limited to terms relevant to our initial efforts. This document will be updated as meanings alter or new terms are needed. As the list grows it may be divided into sections each relevant to different components of the system. We prefer to think of it as a logical description rather than a dictionary. The glossary is written only with the EVLA in mind. It should sometime be read with the future NMA and VLBA operations in mind to see what changes are needed.

Ken Sowinski writes:

> I have taken bits and pieces of various existing documents and have made a first pass at reconciling the disparate definitions. Terms in the definitions which themselves appear in the glossary are written in **bold face**. The term “observation” has been used to mean too many things in this discussion. I suggest that it be left free as a generic word whose meaing is obvious (one hopes) in context. Usually it will mean something like “session”. There has been much confusion over the term used to describe the components of a “scan”. I concur with the usage of the SSR which calls this a “record”.

Proposal:

A request to NRAO for telescope time, including scientific justification, target sources, telescope setup, etc. A proposal is reviewed by referees and if approved is promoted to one or more **projects**.

Project:

One or more projects are derived from an approved **proposal**.

Program:

Within each **project**, observations in different configurations are each assigned to a separate program. Each program has associated with it an observing description (a **program block**), which describes (in greater detail than is available in the **proposal**) how the observations are to be completed for the program.

Program Block (PB):

The set of **scheduling blocks** derived from a single **program** that are to be submitted to the Scheduling Tool for queuing at the same time. An observer would normally create these in a single run of the Observing Tool.

Scheduling Block (SB):

The minimum scheduling unit. If a scheduling block cannot be completed in its entirety, it must be rescheduled. It may not be resumed in the middle. The scheduling block consists of an **observe script** and information for the scheduling tool to schedule it on the array in an appropriate fashion.

Observe Script:

A script in the language of the **real-time** computer system directing the EVLA or a **subarray** thereof to collect data with a specified receiver setup, phase tracking center, and antenna pointing. The EVLA is operated by having the Scheduling Tool, human operator, or observer pass an observe script to the **real-time** computer system. See **subarray** for more about observe scripts and subarrays.

Observed Block:

A succesful execution of the **observe script** associated with a **scheduling block** produces an observed block. There will be one for each execution of a **scheduling block**.

Observing Session:

The time contiguous execution of one or more **scheduling blocks** associated with a single **program**.

Observation:

This term has been used both in the sense of **record** and **observing session**. We should not give any special meaning to this word and understand that when used its meaning is to be derived from context.

Scan:

The scan is the lowest level object normally used by an observer. It is a sequence of one or more **records** that share a single goal. A scan is usually a single **record** of a target source or a calibrator, but, for instance, pointing and focus scans involve a pattern of **records**. This is usually the minimum unit seen by the observing tool.

Record:

A Record is the minimal amount of data taking that can be commanded within an **observe script**. One or more **integrations** comprise a record. An example of a record is a single pointing within a holography **scan**.

Integration:

An Integration is the basic unit of **astronomical data** presented to the **archive**. It is the average of a set of **correlator dumps**.

Correlator Dump:

A Correlator Dump corresponds to the minimum available integration time output from the correlator hardware. One or more of these are averaged to form an **integration**. The minimum collection of data subject to software manipulation.

Pipeline:

A sequence of data reduction operations performed according to a script present at the initiation of Pipeline operation.

Real-time System:

The portion of the EVLA software that understands an **observe script** and actually actually controls the EVLA at the hardware level.

On-line System:

The portion of the EVLA software which directly interfaces to the **real-time** system software, namely the Scheduling Tool, the Archive Tool, the Real-time Calibrator Analysis Tool, the quick-look Pipeline Tool, the Astronomer's What's Up Screen, and any other required software (for instance, that required to do the interfacing).

Astronomy Data:

Whatever is in the telescope's standard raw output data set. For the EVLA it is correlations, various calibration data (sys/cal, etc) flags, odds and ends, and **meta-data**.

Ancillary Data:

Data other than **astronomy data** relevant to the time range of interest, such as operator logs, monitor data, open work orders and known software problems, slowly changing instrumental parameters (e.g. VLA baselines and pointing model parameters), ionospheric data extracted from the GPS network, etc.

Meta-data:

A description of a telescope setup under which **astronomy data** was obtained, containing, but not limited to, the relevant time range, source parameters, receiver setup parameters (especially LO frequencies, and the switch settings used to select bands and filters), the observing proposal in aid of which the observation is made, the observing procedures followed.

Archive:

A permanent collection of all EVLA data and parameters available for access and retrieval. To be distinguished from the physical medium or media where the data resides.

Measurement Set:

AIPS++ format for **astronomy data** from a telescope. *This is a candidate for the format of data communicated between the Correlator Backend and the* **archive***.*

Subarray:

A collection of one or more antennas which are producing data which makes sense to correlate. We distinguish two kinds of subarrays: administrative and astronomical. Administrative subarrays are defined by the operations staff and apportion collections of antennas to independent observers. Astronomical subarrays are defined by the observer from the collection of antennas that were assigned. It is expected that subarrays can be independently configured with no arbitrary restrictions.

B References

[SSR]
EVLA e2e Science Requirements, April 15, 2003

[Morgan]
System Requirements Specification, EVLA Software System, August 22, 2003

[e2e]
e2e Project Book, August 9, 2002

[CBE]
EVLA Correlator Backend, Requirements and Functional Specification, Sep 23, 2003