*10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control Systems. Geneva, 10 - 14 Oct 2005, WE3A.3-6O (2005)*

# The ALMA Telescope Control System

A. Farris, R. Marson, J. Kern
*National Radio Astronomy Observatory, P.O. Box O, Socorro, New Mexico, USA*

## ABSTRACT

The Atacama Large Millimeter Array (ALMA) is a joint project between North America, Europe and Japan. ALMA is an aperture synthesis radio telescope consisting of 50 12-meter antennas located at an elevation of 5,000 meters in Llano de Chajnantor, Chile. These antennas will operate at frequencies ranging from 31.3 GHz to 950 GHz. The antennas can be moved and placed in different configurations, with baselines between the antennas varying from 150 meters to 20 km. The 50 antennas are supplemented by sixteen additional ones, known as the ALMA Compact Array (ACA): 12 7-meter antennas and 4 12-meter antennas.

The ALMA control system will consist of over 70 computers separated by distances of over 20 km. Two aspects of the system are apparent: its distributed nature and its need to accurately synchronize events across many computers separated by large distances. In this paper we describe key features of the architecture of the ALMA Control System, focusing on its properties as a distributed system and on the mechanisms employed to achieve its time synchronization goals.

This control system is a distributed system that uses the ALMA Common Software (ACS) as a middleware system layered on top of CORBA. The architecture of the control system extensively employs the component/container model in ACS. In addition, the use of CORBA allows us to employ Java in the higher levels of the control system, leaving C++ to the lower time-critical levels. Python as a scripting language is used by astronomers, to craft standard observing programs, and engineers, in a testing and debugging mode.

Key to the concept of an aperture synthesis telescope is a special purpose hardware system known as a correlator, responsible for making various delay model corrections and correlating the signals from the antennas. There are two correlators in ALMA, one for the array of 50 antennas and one for the ACA. This entire system operates under a control system that must synchronize events across the entire system to an accuracy of better than one microsecond in order to achieve its scientific objectives.

Time synchronization is accomplished by sending precisely timed electronic signals, which are derived from a Maser, to all hardware that requires accurate time synchronization. The control system preloads time-critical commands and monitor requests that are triggered by these timing signals. Propagation delay times to remote hardware must be taken into account. The slowest of these timing signals, the 48ms timing event, is also connected to all real-time computers. This timing event is used for synchronizing time internal to the array to external time standards. It is part of a hierarchy of timing signals that, in some cases, measures time to subnanosecond accuracy.

## THE ALMA PROJECT

The science goals of ALMA require sub-millimeter wavelengths and an angular resolution of 10 milliarcseconds. In this frequency range water vapor in the earth's atmosphere is a major factor in attenuating the incoming signals. Observing at the highest frequencies in this region is possible only under comparatively rare atmospheric conditions, which is the main reason for choosing the ALMA site: to increase the likelihood that these rare conditions will occur. The ALMA site has been characterized as the driest place on earth.

The antennas themselves, weather stations, the two correlators and their computer interfaces, LO generation hardware, time keeping hardware, and the related ARTM (Array Real-Time Machine) computer are all located at the 5000m site referred to as the AOS (Array Operations Site). This site is connected via Gigabit fiber links to the OSF (Operation Support Facility), located near San Pedro de Atacama at an altitude of 2,800 meters. In addition, science operations will be located at the Santiago Central Office (SCO). Test antennas currently exist at the site of the National Radio Astronomy

Observatory's VLA in Socorro, New Mexico. The first production antenna will be delivered to the site in Chile in approximately 15 months. Additional details about the ALMA project may be found in [1].

## PHYSICAL ARCHITECTURE

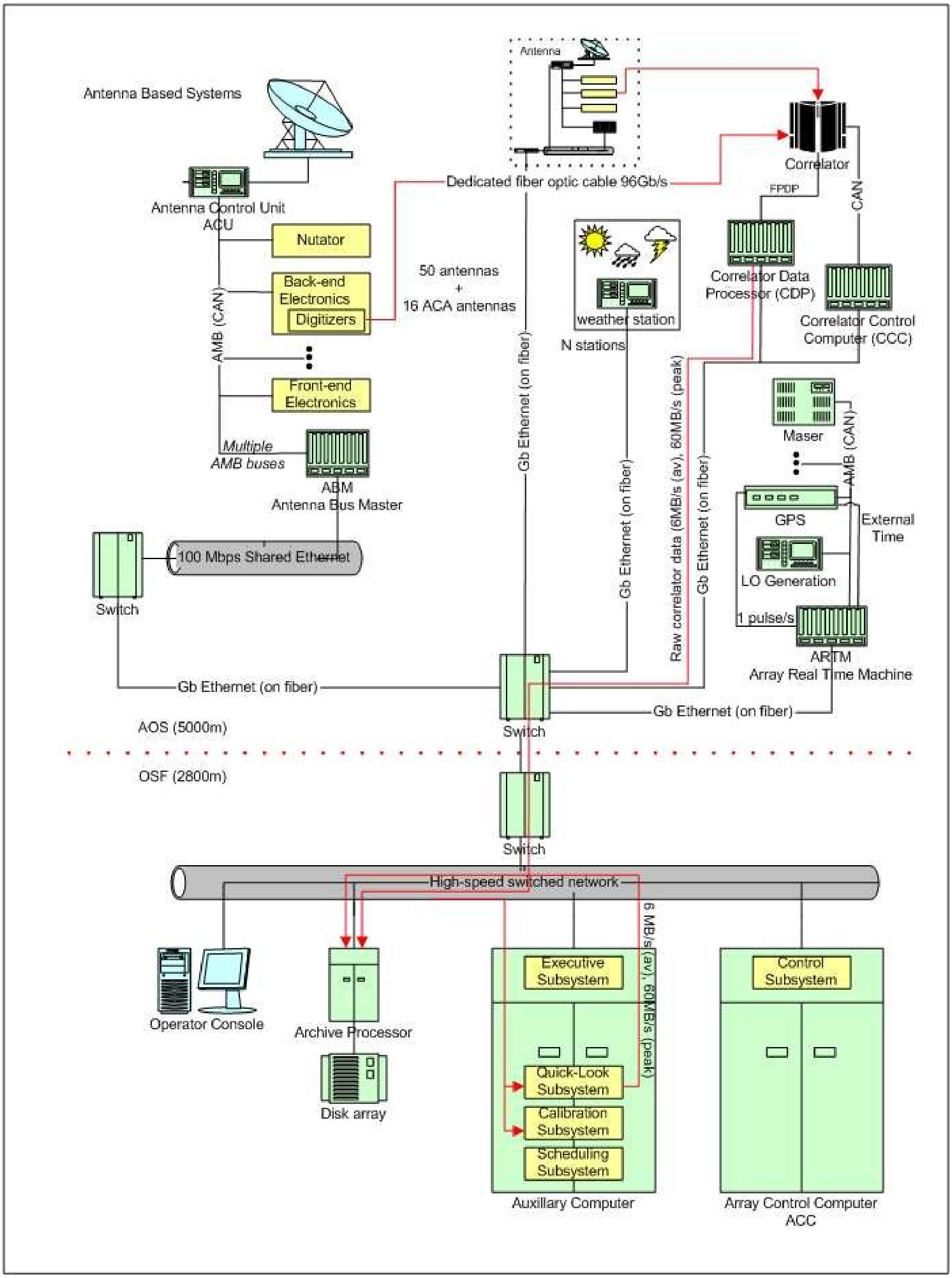

Figure 1 ALMA Physical Architecture

Figure 1 describes the physical architecture of the ALMA system – the layout of principal computers and networks, as well as major hardware subsystems and data flows needed to support science observations. This diagram is intended to portray what is fundamental in understanding the physical architecture of ALMA. It portrays structure at a fairly high level; but it does not distort the underlying details. Only those systems that are essential to supporting science observations are shown. The antennas of the ACA function as shown; only the second correlator to which they are attached is omitted.

At each antenna, the Antenna Bus Master (ABM), a single computer, monitors and controls all the hardware devices in each antenna. This includes equipment ranging from the antenna servos, receivers and data samplers down to the power supplies. This computer utilizes 5 ALMA Monitor Bus's (AMB) channels to communicate with these devices. The AMB is an extension and restriction of the Controller Area Network (CAN) bus [2]. The CAN bus is an industry standard, multi-drop serial connection that uses twisted pair cabling. We use it at the maximum bit rate of 1Mbit/sec and the maximum cable length at this data rate is 35m. We impose a master/slave protocol to ensure that the bus behaves in a deterministic fashion.

Much of the data from each antenna is collected via the ABM and routed, via a gigabit Ethernet, to the central coordinating computers in the control system. However, the high-rate astronomical data collected by the receivers and digitized by the samplers is not routed through the ABM and the control system. Instead it is sent directly to the correlator via specialized mechanisms on fiber optic cables, completely outside the computer system.

The correlator subsystem consists of a real-time Correlator Control Computer (CCC) and a Beowulf cluster of computers known as the Correlator Data Processing computer (CDP). The software of the correlator subsystem runs on both the CCC and the CDP. The raw, correlated output data is routed directly to the archive and to the calibration and quick-look pipeline systems.

There will be a number of weather stations connected to the control system via Ethernet. In addition the ALMA system will deploy one other computer within 40m of central devices, the Array Real Time Machine (ARTM). The ARTM will monitor and control central devices such as the local oscillator generator, GPS and maser. It maintains the time standard for all computers in ALMA.

All of the above computers will operate at the AOS at 5000m. They will be diskless, passively cooled, and will boot and load their software over the network from central computers at the Operations Support Facility (OSF), at 2800m. These computers will run a real-time operating system and use the deterministic AMB, to guarantee they can monitor and control hardware on a fixed time schedule.

The OSF will contain the Array Control Computer (ACC) on which the central portions of the control system will operate. Only the real-time portions of the control system run in the computers deployed at 5000m. The archive processor also operates at the OSF and is essential to the operation of the control system. Figure 1 also shows the executive, calibration, quick-look pipeline, and scheduling subsystems running at the OSF. Strictly speaking, the scheduling subsystem and operator's console are not needed once an observation has started. Similarly, the executive subsystem is not needed to support an observation. However, both scheduling and the operator are essential in starting the observation and the executive detects and stores any anomalous events. In addition, science observations may continue without the quick-look pipeline. However, this pipeline is a normal part of operating the ALMA system and its results are stored in the archive as the observation progresses.

## SOFTWARE ARCHITECTURE

The purpose of the control subsystem is to monitor and command all of the hardware in the ALMA telescope. All on-line hardware is continuously monitored and much of this monitor data is stored permanently in the archive. Due to its geographical distribution and the fact that we use multiple computer languages, we rely on real-time CORBA as the basic mechanism to achieve distributed processing concepts. Our reliance on CORBA is wrapped in middleware called ACS (ALMA Common Software) [4]. The component/container model in ACS is the major feature that allows us to manage and access objects via CORBA. All major software modules (a collection of classes) are components that run under the control of a container.

The Control system uses Java, C++, and Python containers. Every antenna in ALMA has its own container that is run on its ABM computer. Each hardware device on the antenna is represented as a component in that antenna's container. In addition, the antenna itself is a component that is commanded and controlled by the Master Component that controls the entire ALMA array.

Components in ACS may also define properties. These may be hardware device monitor points such as the value of a voltage, temperature, frequency setting, or a state variable (busy, idle, error, etc.). ACS provides facilities for monitoring properties based on time intervals or other logical criteria (such as when value changes by a certain amount). Values of properties are gathered and stored permanently in the archive. This is the basic mechanism by which the Control system stores its monitor data into the archive.

In ACS, notification channels provide asynchronous communication across the ALMA network. Based on the publish/subscribe model and the notification channel concepts in CORBA, any subsystem may create a notification channel and publish events (in the form of IDL structures) and any interested client may subscribe. For example, this is the mechanism that is used to inform other subsystems that a scan (a unit of science data taking) has started and stopped. Control uses two notification channels: an "external" one for communication with subsystems outside of Control and an "internal" one that is used to command antennas and other real-time components within the Control system.

The ALMA software system embraces an end-to-end concept. The major subsystems are depicted in the following. First, an observer creates an observing project using the Observation Preparation subsystem, which breaks the project into scheduling blocks, and stores it in the Archive. Then, the Scheduling subsystem gets project definitions and scheduling blocks from the Archive, dispatching them to the Control system to be executed. The Control subsystem executes a scheduling block by commanding the Correlator. This results in raw data and meta-data being stored in the archive and being made available to the real-time Telescope Calibration and Quick-Look Pipeline. The completion status of scheduling blocks is monitored by the Scheduling subsystem, which, in turn, starts the Science Data Reduction Pipeline at appropriate times. That Science Pipeline generates calibrated data products that are stored in the Archive. The Scheduling subsystem also monitors the completion of the science data reduction and informs the principal investigator when completed project data are available. This entire process is started, stopped, and continuously monitored and controlled by the Executive subsystem. The detailed architecture of the ALMA software system is described in [3]. Within this context, Control, Correlator, Archive, Telescope Calibration and the Quick-Look Pipeline run in real-time.

Figure 2 depicts the run-time structure of the Control subsystem. The Control subsystem operates under the control of a Master Component, which implements the following actions:

- Start/stop all resources and components – initiated by the Executive subsystem.
- Monitor the state of all devices – a continuous activity.
- Create/destroy automatic array – initiated by the Scheduling subsystem.
- Create/destroy a manual array – initiated by either the Scheduling or Executive subsystems.
- Place an antenna (or, parts of an antenna) online/offline – initiated by the Executive subsystem via the telescope operator interface.
- Publish a change of state event – all major changes of state in the Control system are published as events on Control's external notification channel.
- Read and update the telescope configuration database – all major changes of state in the Control system are recorded in the telescope configuration database.
- Analyze and respond to errors in subordinate components – any errors in subordinate components that those components cannot deal with are reported to the Control Master Component.

Antennas in the ALMA telescope may be grouped into independent subsets, called arrays. For example, a common way of using ALMA will be to have one array that includes the 50 "main" antennas and a second array of the ACA. Arrays in the Control subsystem are created dynamically and are either automatic or manual. Automatic arrays are the "normal" mode of operation and are used to execute scheduling blocks of approved scientific projects. An automatic array utilizes a script executor that executes an observing script, a component that monitors the state of execution, a delay server that computes the relative geometric delay of the antennas and a data capture component that creates the science meta-data that is stored in the archive. Manual arrays are used by engineers to debug problems or by staff astronomers to develop new modes of observing. All features of the allocated antenna hardware are accessible to a manual array. There is a considerable variation in the functional capability of a manual array – it may be configured to merely allow engineering diagnostics and store no data in the archive, or it may be configured with full delay server and data capture functionality.

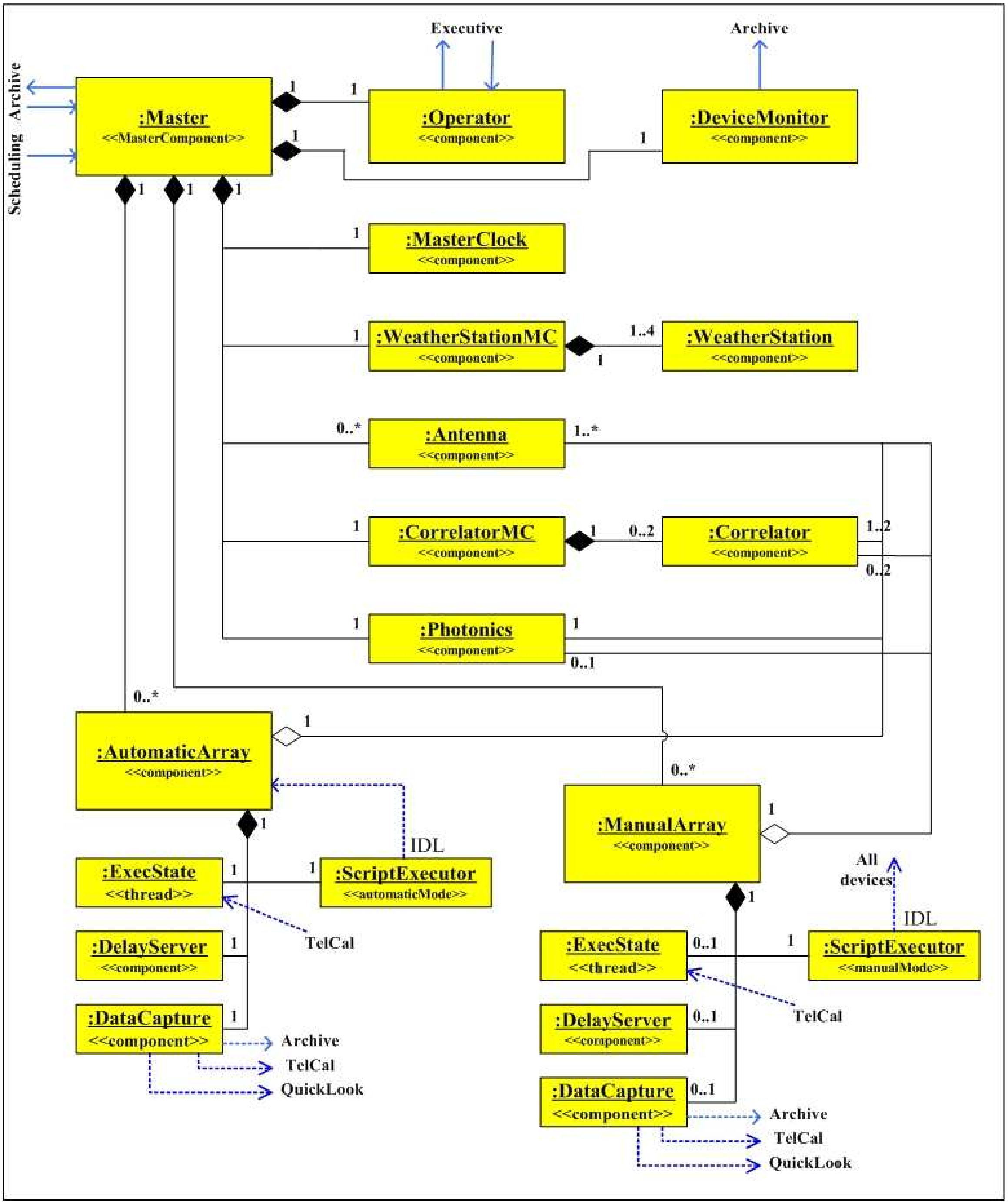

Figure 2 Run-Time Structure of Control

# TIME AND SYNCHRONIZATION

It is apparent that precise timing and synchronization of events across the entire array present major challenges to the Control system. Within the ALMA computer network, computers are divided into real-time and non-real-time. Non-real-time computers use the Network Time Protocol (NTP). However, NTP is too inaccurate for real-time computers within the ALMA network.

The Control system relies on a combination of hardware and software, synthesized into a unit known as the "Master Clock", to coordinate activities across the entire array. The hardware parts of the Master Clock consist of a maser as the fundamental time standard, a GPS to link the maser to external time (TAI), and a central reference generator to generate and distribute, across the array, the fundamental timing signals. This hardware is connected to the ARTM computer that coordinates the hardware and synthesizes the time reference that is used by all computers in ALMA. In particular it provides a network time server that is used by all non-real-time computers in the array and a custom IDL interface that is used by the real-time computers.

The Central Reference Generator (CRG) produces, from the Maser output, three precise phase locked signals, a 2GHz signal, a 125MHz signal and a 20.833 periodic pulse, known as the timing event. The timing event (TE) is used to interrupt every real-time computer every 48 milliseconds. This window of time is the basic mechanism used to synchronize events across the array. The first 24 milliseconds of the TE window are used to send real-time commands to the attached hardware. Commands that are executed during this period have a hard real-time deadline and are triggered and activated in the hardware, by the subsequent TE. The other timing signals generated by the CRG are used in hardware to establish time references with a precision of better than 1ns. At the antenna, these signals are used to synchronize local oscillators, fringe rotation offsets, digitizer clocks, data transmission clocks and phase switching. In practice this accuracy will be affected by the extent to which propagation delays can be taken into account.

The method of synchronizing events depicted above may be thought of as "internal" time and is accomplished by coherence among local oscillators. Associating those events with accurate external time is important for observational astronomy: to allow correct real-time evaluation of the ephemeredes of astronomical objects and to allow coordination of observations with other observatories (as in VLBI). Except for initialization, the time keeping protocol is very simple; the ARTM is told the time associated with a specific timing event and it then maintains time by counting TEs and ensuring that its internal clock increments by exactly 48 milliseconds every timing event. The initialization is accomplished by having the ARTM connected to an electronic source of external time – a GPS receiver. The ARTM will get its time initially from this device, using a serial connection. It then counts pulses to maintain its time. The other real-time computers maintain their own time by also counting TE's. When they are booted, one of the first things they do is request, via a CORBA function call, the time at the next TE from a component running on the ARTM,. The response to this request must be received within 48ms to ensure that the correct timing event is used. The remote computer times the length of the transaction and, if it is longer than 48 milliseconds, tries again (for a preset number of times). The remote computer always knows if it has succeeded in synchronizing its time. If it fails, it generates an appropriate error condition. Likewise, all real-time computers can detect when they have missed a TE. This signals an error that results in a command to resynchronize the local clock via a "getTimeAtNextTE" call.

One might describe this design as accomplishing synchronization via the coherence of the local oscillators and the stability of the reference distribution system, while maintaining external time in software.

# REFERENCES

[1] "The ALMA Computing Project – Update and Management Approach," B. Glendenning, G. Raffi, paper MO2.2-1I, ICALEPCS 2005.

[2] "ALMA Monitor and Control Bus Interface Specification," M. Brooks, L. D'Addario, ALMA document ALMA-70.35.10.03-001-A-SPE, 2001-09-07.

[3] "The ALMA Software System," J. Schwarz, H. Sommer, A. Farris, Astronomical Data Analysis Software and Systems, XIII, ASP Conference Series, Vol. 314, 2004, F. Ochsenbein, M. Allen, and D. Egret, eds., pp. 634 – 652.

[4] "The ALMA Common Software ACS Status and Developments," G.Chiozzi, A.Caproni, B.Jeram, H.Sommer, M.Plesko, M.Sekoranja, K.Zagar, D.Fugate, S.Harrington, R.Cirami, P.Di Marcantonio, paper WE2.4-6I, ICALEPCS 2005.

[5] "Timing and Synchronization," L. D'Addario, ALMA Memo No. 298, 2000-Mar-09.