Journal of Physics: Conference Series **513** (2014) 012030 doi:10.1088/1742-6596/513/1/012030

# The readout and control system of the mid-size telescope prototype of the Cherenkov Telescope Array

**I Oya[1], O Anguner[1], B Behera[2], E Birsin[1], M Fuessling[3], D Melkumyan[2], T Schmidt[2], U Schwanke[1], R Sternberger[2], P Wegner[2], S Wiesand[2] for the CTA Consortium[4]**

[1]Institut für Physik, Humboldt-Universität zu Berlin, Newtonstrasse 15, D-12489 Berlin, Germany
[2]Deutsches Elektronen-Synchotron, DESY, Platanenallee 6, D-15738 Zeuthen, Germany
[3]Institut für Physik und Astronomie, Universität Potsdam, Karl-Liebknecht-Strasse 24/25, D 14476 Potsdam, Germany
[4]see http://www.cta-observatory.org/ for full author and affiliation list

E-mail: oya@physik.hu-berlin.de

**Abstract.** The Cherenkov Telescope Array (CTA) is one of the major ground-based astronomy projects being pursued and will be the largest facility for ground-based γ-ray observations ever built. CTA will consist of two arrays: one in the Northern hemisphere composed of about 20 telescopes, and the other one in the Southern hemisphere composed of about 100 telescopes, both arrays containing telescopes of different type and size. A prototype for the Mid-Size Telescope (MST) with a diameter of 12 m has been installed in Berlin and is currently being commissioned. This prototype is composed of a mechanical structure, a drive system and mirror facets mounted with powered actuators to enable active control. Five Charge-Coupled Device (CCD) cameras, and a wide set of sensors allow the evaluation of the performance of the instrument. The design of the control software is following concepts and tools under evaluation within the CTA consortium in order to provide a realistic test-bed for the middleware: 1) The readout and control system for the MST prototype is implemented with the Atacama Large Millimeter/submillimeter Array (ALMA) Common Software (ACS) distributed control middleware; 2) the OPen Connectivity-Unified Architecture (OPC UA) is used for hardware access; 3) the document oriented MongoDB database is used for an efficient storage of CCD images, logging and alarm information: and 4) MySQL and MongoDB databases are used for archiving the slow control monitoring data and for storing the operation configuration parameters. In this contribution, the details of the implementation of the control system for the MST prototype telescope are described.

## 1. Introduction

Very High Energy (VHE, energies > 100 GeV) γ-rays are produced in astronomical sources like pulsars, pulsar wind nebulae, supernova remnants and active galactic nuclei. Thanks to the success of Imaging Atmospheric Cherenkov Telescope (IACT) instruments like H.E.S.S. [1], MAGIC [2] and


Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

VERITAS [3] the field has experienced a major boost during the recent years, with more than 100 sources already established as VHE emitters. The technique used by IACTs is based on the fact that VHE γ-rays interact in the atmosphere producing electromagnetic showers, the charged components of which generate Cherenkov light in a narrow cone. An IACT consists of a reflector that concentrates the Cherenkov light in a camera composed of sensitive photo-detectors, usually photomultiplier tubes (PMTs). In order to fulfill its scientific requirements, CTA will comprise two arrays, one in the Northern hemisphere composed of about 20 telescopes, and the other one in the Southern hemisphere composed of about 100 telescopes, both arrays containing telescopes of different type and size. A few large size telescopes (LST), with dishes of 23 m diameter, are designed to achieve a good sensitivity in the energy domain from a few tens of GeV up to about a hundred GeV. In order to achieve a good sensitivity at energies above a few TeV a large number of small-size telescopes (SSTs), will be deployed. For good sensitivity at core energies (around 100 GeV to 1 TeV) a considerable number of MSTs, of Davies Cotton [4] design and 10-12 m diameter dish each, will be used. Additionally, between 20 to 30 extra MST telescopes of an alternative design using a Schwarzschild-Couder [5] layout will be deployed. By extrapolating the distribution of known VHE objects, it is expected that CTA will detect around 1000 new objects, including new source type detections and key contributions to fundamental physics. Details of the science motivation, design concepts and expected performance of CTA are provided in [6].

## 2. The CTA array control challenge

The design and implementation of the array control software for CTA constitutes a remarkable challenge. First, it will have to monitor and control dozens of telescope units, of at least 4 different types (with rather different designs), plus auxiliary complex devices like lidars and optical telescopes, deployed in an installation an order of magnitude larger than any current existing IACT installation. Secondly, different modes of operation have to be supported; some of the observations will be pointed observations of specific astronomical targets, while systematic scans of certain regions of the sky like the Galactic plane are also foreseen. During some of the operations the whole array will be used, while during others the array will be fragmented in several *sub-arrays* operating concurrently. In multi-wavelength campaigns, observations will be performed in coordination with instruments operating in other wavelengths. Finally, CTA will be ready to automatically react to alerts issued by other ground or satellite installations immediately proceeding to observe a specific astronomical object (*target of opportunity observations*), and, conversely, provide automatic alerts to other installations when an interesting situation is detected by the real-time analysis of CTA data. All these requirements motivate a careful selection and evaluation of technologies for the CTA array control software.

Due to the similar level of complexity and requirements of ALMA [7] and CTA, the ALMA Control Software [8] (ACS) is currently being considered by the CTA consortium members to serve as the CTA array control middleware (see [9, 10] for more details). In addition, in order to standardize the interface to the hardware, a lower level software layer provided by the OPen Connectivity-Unified Architecture (OPC UA) is under evaluation by the CTA consortium.

## 3. The Mid-Size telescope prototype

A prototype for one of the design concepts of the MST has been deployed in Berlin-Adlershof and it is currently in commissioning phase. The MST prototype has a Davies-Cotton type reflector with a diameter of 12 m, and a focal length of 16 m. The main goal of this prototype is to test the design of the mechanical structure and drive system together with other prototype instruments like the Active Mirror Control (AMC). A “dummy” camera of 2.5 tons is installed to measure the structural effects of

a real PMT camera. The prototype will also contain five Charged-Coupled Device (CCD) cameras and a Weather Station (WS), plus non-optical sensors designed to test the mechanical structure.

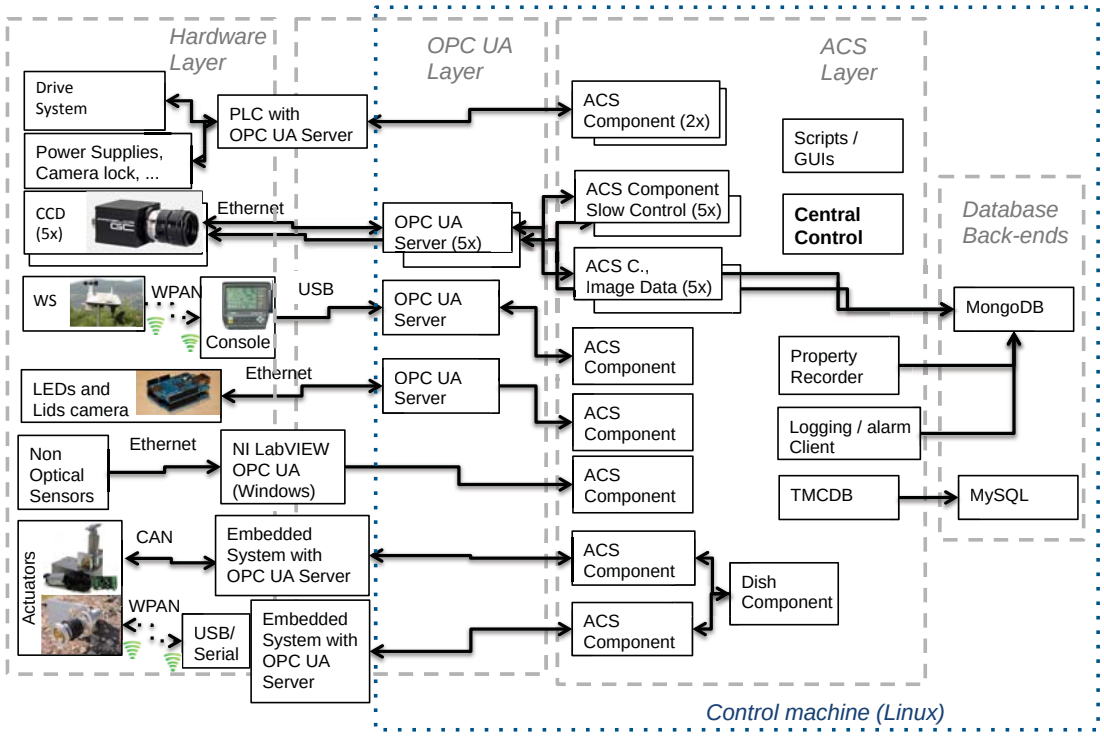



**Figure 1.** The MST prototype and instrumentation for control and readout. See text for details.

## 4. The control software of the Mid-Size telescope

The MST prototype constitutes a very good opportunity inside the CTA consortium to prototype the integration of a considerable number of hardware devices within a control software framework, and thus evaluate and exercise the ACS and OPC UA software technologies mentioned in section 2. In particular, OPC UA servers for hardware access are used to interface most of the devices, and ACS components are used for the higher level functionality. In the context of the CTA MST prototype, most of the OPC UA servers are being modelled using the Prosys OPC UA Java Software Development Kit (SDK). Besides the servers created with the Prosys SDKs, additional OPC UA servers built with other tools are also used and will be discussed later in the text. The standard ACS access method (a Java DevIO class (see [11, 12] for details) is used for most of the communication between ACS components and OPC UA servers, complemented with Prosys OPC UA SDK for method invocation and event/alarm transmission.

The MST control software is deployed in a Dell PowerEdge R720xd machine with 2 CPUs (12 CPU cores), 128GB main memory and more than 6TB local disk space. Additionally, a software development, integration and test environment for the MST control software exists in DESY Zeuthen, comprising two Dell PowerEdge R510 machines with 12 Nehalem Cores, 36 GB RAM memory and 10 Gbit Ethernet interface using the same layer-2 switch. The operating system used is Scientific Linux, version 6, which is compatible with RedHat Enterprise 6. The MST team developed an RPM-based installation procedure of 64-bit ACS releases, and it is used to deploy the ACS software in both the control and test environment. An archive for permanent storage of the prototype measurements is

under development, which will also serve as a test bench for the CTA archive. The temporal archive of the acquired data is deployed at the MST control site.

Figure 1 shows an schematic view the main instrumentation and software components of the telescope control, whereas Fig. 2 shows the data flow. In the remainder of this section, the prototype telescope devices will be described, together with the solutions used for their steering and readout.

### *4.1. The CCD cameras*

A crucial tool to check different aspects of the MST prototype will be a set of five CCD cameras. These cameras will enable the measurement and correction of the pointing accuracy of the telescope, testing of the structural design of the prototype and, with the help of a WS, the investigation of the effect of environmental conditions. Additionally, the CCDs will be used to perform mirror adjustments, and will allow for measurements of the optical point spread function. These CCDs will operate at a rate up to 10 Hz and will generate the largest fraction of data volume of the prototype.

The CCD camera model used is a Prosilica GC 1350, which has a resolution of 1.4 Mpix (1360 x 1024) and a pixel size of 4.65 μm. These CCD cameras are interfaced via a GigE Vision interface (allowing up to 1000 Mbit/s on Gbit Ethernet). The Allied Vision Technologies (AVT) PvAPI SDK allows to control and capture images from GigE Vision CCD cameras in a Linux environment. Java OPC UA servers are used as the low level access to the CCD cameras. On a higher level, the ACS part of the control software consists of several ACS components that use the Java DevIO mechanism for configuring the camera parameters and to obtain images. Additional ACS components are used for storing the CCD images (see Section 4.5). A graphical user interface is used to display image frames in real time. Further details can be found in [12].

### *4.2. The Drive system*

The drive system of the MST prototype is designed to implement the expected operation modes of the CTA telescopes, allowing pointing of the prototype to any position in the sky and to track any astronomical object. The telescope will operate with two motors, one for azimuth and one for elevation. The drive system is composed of 6 motors (2 for azimuth and 4 for elevation) communicating via a Bosch-Rexroth programmable logic controller (PLC). The PLC operates a VxWorks real-time operating system and hosts an OPC UA server. This OPC UA server is accessed from an ACS component by using the OPC UA DevIO mechanism. In order to fine tune the calibration of the drive system, the telescope control system is being used to track astronomical objects and to check the pointing by taking images of the field of view with a CCD camera, which are analysed with the astronomy.net software to find the actual targeted position. This will also allow to determine the telescope *bending model,* that will allow to minimize the effect of the weight of the structure, which depends on the telescope elevation, in the pointing accuracy.

### *4.3. The AMC system*

The design of the CTA telescopes makes use of a tessellated reflector composed of individual mirror facets (see [13] for details). Each individual mirror facet will be attached to a triangular support, with two power actuators and a fixed support (an Active Mirror Control (AMC) unit) which has the functionality to enable perfect mirror alignment. This will allow online re-alignments of the LSTs's reflectors, where the deformation caused by the weight of the telescope will cause misalignments depending on the telescope elevation. It will also allow, on longer timescales, realignments of MSTs and SSTs to be performed. The dish of the MST prototype will be completely covered by a combination of real and dummy mirrors. Several AMC units of two different designs will be installed

at the MST prototype (see [6] for details in the design concepts). One type of unit communicates via XBee radio modules, creating a Wireless Personal Area Network (WPAN) that is accessed via a XBee receiver connected to an embedded computer via USB or RS-232 serial interface. The other type of unit is interfaced via Controller Area Network Bus (CAN-Bus), accessed via an Ethernet-CAN-Bus gateway. The mirror alignment procedures use the images obtained with the CCD cameras as input, and therefore the higher level component will be synchronized with the operation of the CCDs. The control software of the AMCs is being developed in C++/Java under ACS.

### *4.4. Other devices*

**Weather station:** Attached to a lightning protection pole near the MST prototype, a Davis Vantage Vue WS continuously monitors the weather parameters and allows correlation of the parameters with the behavior of the structure of the MST under changing environmental conditions. The instrument is composed of an outdoor unit communicating via a WPAN with an indoor unit equipped with a RS-232 serial interface. At the lowest software level, an OPC UA server has been implemented with the OPC UA Java SDK, which uses the RXTX library to communicate with the WS data-logger. At a higher level, the data from the WS is accessed via a Java ACS component.

**LEDs and lids camera:** In order to emulate the weight and some of the functionalities of a real IACT camera, a camera "dummy" is installed in the focal plane of the MST. The camera can open and close protective lids, power LEDs (for pointing calibration purposes, together with the CCD camera), and provides temperature and accelerometer sensors. The device is controlled by an Arduino Mega micro-controller, and interfaced to the central control via a OPC UA server and an ACS component, both implemented in Java.

**Structure monitoring sensors:** The telescope structure is monitored via a set of sensors that measure the acceleration, inclination, temperature, tension and material stretch. These sensors are read out with a National Instruments (NI) 9025 input module with 32 channels data acquisition system. The readout software was developed in LabView, which includes an embedded OPC UA server. The telescope control software will interact with these sensors with an ACS component (under development), via the OPC UA server.

### *4.5. Data persistency and archives*

In the scope of the MST prototype telescope, four different tasks require the use of archives. These tasks are similar, but on a much smaller scale, to those that will be required for the CTA array control, therefore allowing an early evaluation of technologies, as well as their integration in the control middleware. These are:

**Configuration data:** By default, the ACS system uses a Configuration Database (CDB) to configure the component parameters and deployment. The default implementation of CDB consists of a set of XML files parsed against some corresponding XSD files. ACS also provides an alternative implementation of the CDB that uses a relational database such as HSQLDB or Oracle DB to store this information. The database-oriented implementation has been extended to run at the MST prototype with MySQL.

**Monitoring data:** Each ACS component has a number of properties. Properties are used to monitor (and eventually control) the corresponding entity such as physical devices. A generic property monitoring system has been developed, that consists of a front-end, which automatically detects any ACS component and continuously acquires its property data plus a back-end that stores the data. Back-

ends using MongoDB and the ACS logging system are already implemented; alternative back-ends using MySQL and HDF5 files are planned.

**Log messages and alarm information:** All ACS log messages and alarms are transmitted to central services that provide interfaces for consumers of these data. Two consumers (one for logs and another for alarms) were realized for the MST control system. They store all data in MongoDB, thus providing powerful tools for querying and filtering logs and alarms, and also avoiding the risk of filling up the disk with excessive log information.

**CCD images:** CCD images represent the biggest part of data at the MST prototype. They are stored with GridFS, which is an abstraction layer on top of MongoDB to store and retrieve the CCD pixel data content plus the meta-data. The CCD software contains a specialized CCD listener that receives image data from the CCD image server and writes it to GridFS.

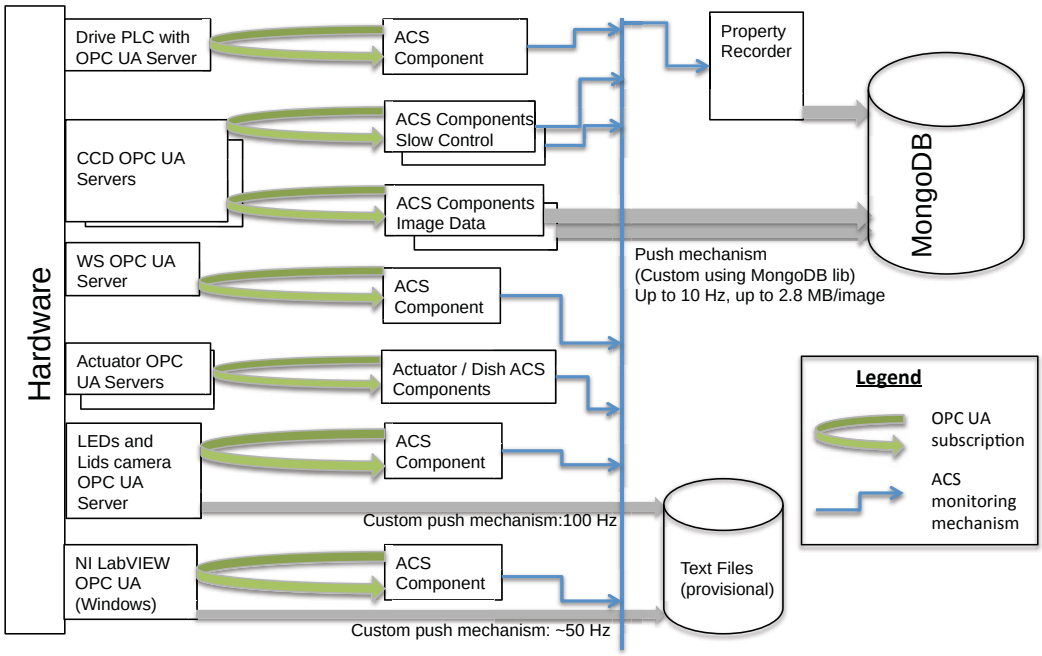



**Figure 2.** The MST prototype data flow, dominated by the CCD data. The standard OPC UA subscription is used to synchronize the slow control data in the ACS components. The ACS monitoring mechanism consists of both a time-triggered and value change-triggered polling.

### *4.6. First integrated operations*

In an initial test, the ACS-based software has been successfully used to move the telescope while acquiring data simultaneously with the operation of the three installed CCD cameras, the LED and lid camera, and the WS. In another operation, one of the CCD cameras was used to acquire images from stars in a clear night while the drive system was tracking a sky position at sidereal speed. The processing of the obtained images, stored in the MongoDB database, allowed the extraction of the pointing direction of the CCD camera. (See [14] for further details)

## 5. Conclusions and outlook

The MST prototype is providing a realistic test bench scenario for the control software of the CTA array. Equipped with complex systems like the drive system, CCD cameras and the AMC, the MST prototype allows the evaluation of the ACS and OPC UA frameworks while the hardware design concepts are being tested. These activities serve as an important input for the CTA array control software developers. At the time of writing this document, 7 OPC UA servers and 24 ACS components have been deployed at the prototype site, with more to come. First integrated operations, using several hardware devices, have been performed.

In addition, the MST prototype team is currently designing a set of GUIs based on web browser technologies for displaying the slow control data and to serve as the central control interface for the telescope. Among others, *Grails* and *Django* are being considered. This will allow to prototype and gain experience with these tools for the CTA array control implementation.

## 6. References

[1] Aharonian F et al (H.E.S.S. Collab.) 2006 Observations of the Crab nebula with HESS *A&A* **457** 899
[2] Albert J et al (MAGIC Collab.) 2008 VHE-Ray Observation of the Crab Nebula and its Pulsar with the MAGIC Telescope *ApJ* **674** 1037
[3] Acciari V A et al (VERITAS Collab.) 2008 VERITAS Observations of the γ-Ray Binary LS I +61 303 *ApJ* **679** 1427
[4] Davies J M and Cotton E S 1957 Design of the quartermaster solar furnace *Solar Energy* **1** 16
[5] J. Rouselle, J et al 2013 Schwarzschild-Couder telescope for the Cherenkov Telescope Array: Development of the Optical System. Proc. of the 33rd International Cosmic Ray Conference Rio de Janeiro, Brazil, 2013
[6] CTA Consortium 2011 Design concepts for the Cherenkov Telescope Array *Experimental Astronomy* **32** 193
[7] Wootten A and Thomson A R 2009 The Atacama Large Millimeter/submillimeter Array *IEEE Proc*. **97** 1463
[8] Chiozzi G et al 2004 The ALMA Common software: a developer friendly CORBA based framework *SPIE Proc*. **5496** 205
[9] Oya I et al 2011 A readout and control system for a CTA prototype telescope Proc. Of ICALEPCS 2011 Grenoble, France, October 2011
[10] Wegner P et al 2012 Simultaneous Operation and Control of about 100 Telescopes for the Cherenkov Telescope Array *J. Phys.: Conf. Ser*. **396** 012052
[11] Di Marcantonio P et al 2011 Evaluation of software and electronics technologies for the control of the E-ELT instruments: a case study Proc. of ICALEPCS 2011 Grenoble, France, October 2011
[12] Oya I et al 2012 Evaluating the control software for CTA in a medium size telescope prototype *J. Phys.: Conf. Ser*. **396** 012037
[13] Foster A et al 2011 Mirror development for CTA Proc. of the 32nd International Cosmic Ray Conference Beijing, China, 2011
[14] Wegner P et al 2013 Status of the ACS-based Control System of the Mid-sized Telescope Prototype for the Cherenkov Telescope Array (CTA) Proc. of ICALEPCS 2013 San Francisco, USA, October 2013

## Acknowledgements

We thank the whole MST prototype group at Humboldt University Berlin and DESY Zeuthen, engineers and scientists, for their excellent work. We gratefully acknowledge support from the agencies and organisations listed at the following URL: http://www.cta-observatory.org/?q=node/22