A READOUT AND CONTROL SYSTEM FOR A CTA PROTOTYPE TELESCOPE

I. Oya, U. Schwanke,
Institut für Physik, Humboldt-Universität zu Berlin, Berlin, Germany
B. Behera, D Melkumyan, T. Schmidt, P. Wegner, S. Wiesand, M. Winde,
Deutsches Elektronen-Synchotron, DESY, Zeuthen, Germany
for the CTA Consortium*

Abstract

CTA (Cherenkov Telescope Array) is an initiative to build the next generation ground-based γ-ray instrument. The CTA array will allow studies in the very-high-energy domain in the range from a few tens of GeV to more than a hundred TeV, extending the existing energy coverage and increasing by a factor 10 the sensitivity compared to current installations, while enhancing other aspects like angular and energy resolution. These goals require the use of at least three different sizes of telescopes. CTA will comprise two arrays (one in the Northern hemisphere and one in the Southern hemisphere) for full sky coverage and will be operated as an open observatory.

A prototype for the Medium Size Telescope (MST) type is under development and will be deployed in Berlin by the end of 2011. The MST prototype will consist of the mechanical structure, drive system and active mirror control. Four CCD cameras and a weather station will allow the measurement of the performance of the instrument. The ALMA Common Software (ACS) distributed control framework has been chosen for the implementation of the control system of the prototype. In the present approach, the interface to some of the hardware devices is achieved by using the OPC Unified Architecture (OPC UA). A code-generation framework (ACSCG) has been designed for ACS modeling. In this contribution the progress in the design and implementation of the control system for the CTA MST prototype is described.

THE CTA ARRAY

Very High Energy (VHE, energies $>$ 100 GeV) γ-rays are produced in both galactic and extra-galactic objects like pulsars, pulsar wind nebulae, supernova remnants and active galactic nuclei. In recent years, thanks to the success of Imaging Atmospheric Cherenkov Telescope (IACT) instruments like H.E.S.S., MAGIC and VERITAS the field has experimented a major boost, with more than 100 sources already established as VHE emitters. The technique used by IACTs is based on the fact that VHE γ-rays interacting with the atmosphere produce electromagnetic showers, whose charged components 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). The measured light distribution in the camera of each recorded event allows to reconstruct the arriving direction, energy and type of primary particle which caused the particle shower. In systems of telescopes the individual images can be combined to provide stereoscopic information, enhancing the resolution and background suppression.

The goals of CTA require the use of more than 50 telescopes of three different sizes. 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 the core energies a considerable number of MSTs, of 10-12 m diameter each, will be used.

CTA will comprise two arrays (one in the Northern hemisphere and one in the Southern hemisphere) for full sky coverage and will be operated as an open observatory. By extrapolating the distribution of known VHE objects, it is expected that CTA will detect around 1000 new objects. Details on the science motivation, design concepts and expected performance of CTA are provided in [1].

THE MEDIUM SIZE TELESCOPE PROTOTYPE

A prototype of one of the design concepts of the MST is under development [2] and will be deployed in Berlin (Adlershof Campus) by the end of 2011 or the beginning of 2012. The main goal of this prototype is to test a design of the mechanical structure and drive system, but other prototype instruments like the Active Mirror Control (AMC) units will also be tested. No PMT camera will be installed on this prototype.

The MST prototype has a Davies-Cotton type reflector with diameter of 12 m, and a focal length of 16 m. The design of the telescope structure can be seen in Fig 1. Besides the mechanical structure, the prototype will consist of a dummy camera of 2.5 tons (to resemble the effect of the real PMT camera that will be in the final MST units), the drive system, active mirror control, four Charged-Coupled Device (CCD) cameras and a Weather Station (WS), plus some additional sensors designed to test the behavior of the structure. Time stamps of events will be obtained either via the Network Time Protocol (NTP) or a Global Position System (GPS) receiver. Emulated data sources will be

* see http://www.cta-observatory.org/ for full author and affiliation list

employed to resemble a realistic operation scenario of the prototype. Details of individual devices are provided later in this document. A full size prototype of one-quarter section of the dish has been constructed. The purpose of this prototype is to test the fabrication methods, costs, and to provide a *test bench* for mounting the mirrors.

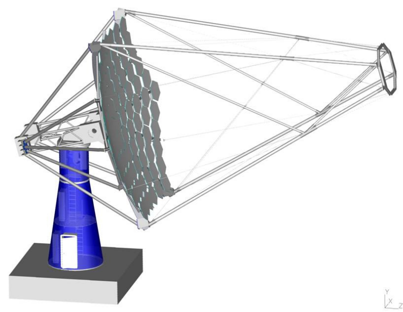

Figure 1: View of the design of the mechanical structure of the MST, composed by a quadrupod (camera support), a dish structure, a counterweight, the "head" (connecting the dish and the tower), and the tower with the foundation.

CONTROL SOFTWARE FOR THE MST PROTOTYPE

The MST prototype is the first opportunity inside the CTA consortium to integrate a considerable number of hardware devices and to exercise the control software. Each individual CTA telescope will be a complex system that has to operate in synchronization with the other telescopes and auxiliary devices. The CTA consortium will allow flexible observation scheduling by including the possibility of fragmenting the arrays in several sub-arrays.

Typically, a control software for an installation of the size of CTA should allow a flexible way of accessing the hardware. Engineers would require expert modes of accessing the devices for testing purposes, while astronomers will require a framework to operate an array composed of several dozens of telescopes, typically without being familiar with the complex details of the devices. The mentioned constrains fit naturally in the distributed computing paradigm, and in particular in the Object Management Group (OMG) CORBA specification, the most successful distributed object platform.

The ALMA Control Software [3] is a framework on top of the operating system that provides a complete environment and structures at the base of application software developments. ACS is meant to be a general system, based on CORBA and the C++, Java and Python languages. The core of ACS is based on a distributed component model, with ACS components implemented as CORBA objects in any of the supported programming languages. ACS Components are the base for high level control entities and for the implementation of devices such as a telescope drive system. ACS provides common CORBA-based services such as logging, error and alarm management, configuration database and lifecycle management, while hiding the complexities of CORBA programming. ACS is developed by the ALMA project and the main responsibility for the subsystem is at European Southern Observatory (ESO).

Due to the similar level of complexity and requirements of ALMA and CTA, ACS is currently being considered by the CTA consortium members to serve as the control middleware. It has been decided to use ACS for controlling the MST prototype because it will provide an excellent platform to test the behavior of ACS with CTA hardware, while serving as a learning platform.

CTA will be composed of a large number of devices that in principle might use very different hardware interfaces (serial lines, Ethernet, CAN-bus, etc.) Large experiments benefit by setting a standard way of accessing the hardware, either by defining a standard hardware interface like in ALMA (CAN-Bus) or by setting a lower level software layer like, for example, OPen Connectivity-Unified Architecture (OPC UA). For CTA it is in general preferred to select devices using Ethernet interfaces, and to set OPC UA servers for those devices requiring a different interface. Some preliminary studies and prototype developments have been taken place at LAPP[1] [4] in order to illustrate the implementation of the OPC UA standard. In particular these implementations concerned some electronic components and related functionalities devoted to the online security control of the H.E.S.S. 2 [5] camera. These studies have confirmed the homogeneous way of connecting electronic boards via the low software layer of the OPC UA and the standard way of integration in the ACS environment (by means of the DevIO mechanism in the ACS terminology). A similar approach has been adopted for the MST prototype: A standard way of deploying OPC UA servers has been investigated, and an ACS access method (a Java DevIO class) is under development with the support of the ACS development team from ESO, complementing the existing C++ DevIO for OPC UA [6].

The ACS components for the control of the MST prototype are being designed using an Unified Modeling Language (UML) based code generator. The UML models are created using MagicDraw UML Standard v.17.0 on Linux, which is used to create the input required by the ACS code generation framework (ACSCG) [7]. From the UML model, ACSCG generates the interface definition language (IDL), configuration database files and Java implementation skeleton. The Java skeleton can be extended to provide an OPC UA client functionality, in order to communicate with an OPC UA server. In order to model the OPC UA servers the OPC UA Java Software Development Kit (SDK) v. 1.1.0-975 is being used.

A software development, integration and test environment has been deployed in DESY, composed by two Dell PowerEdge R510 machines with 12 fast disks, 12 Nehalem

[1] Laboratoire d'Annecy-le-Vieux de Physique de Particules- http://lapp.in2p3.fr/

Cores, 36 GB RAM memory and 10 Gbit Ethernet interface using the same layer level 2 switch. A RPM based installation of 64 bit ACS releases has been developed. 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 DESY data-center will be connected to the MST prototype site, ca. 15 km away, by a broadband connection.

PROTOTYPE INSTRUMENTATION

Drive System

The drive system of the MST prototype is designed to resemble the expected operation modes of the CTA telescopes, allowing pointing of the prototype to any position and to track any astronomical object. The telescope will operate with two motors, one for azimuth and one for elevation. As a first instance to test the drive concept, a test stand was build to evaluate the design (see Fig 2).

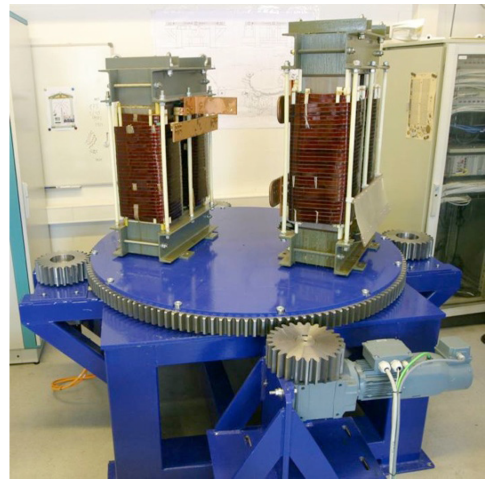

Figure 2: The MST prototype test stand.

The drive system of the prototype is composed, at a lower level, of the control of 6 drives (2 for azimuth and 6 for elevation) communicating via a *Bosch-Rexrot* programmable logic controller (PLC). The PLC operates VxWorks and hosts an OPC server. Earlier versions of the PLC firmware provided an OPC Data Access (DA) specification server, with plans to be replaced with an OPC UA server. As OPC DA is only working for Microsoft Windows environments, and for CTA Linux based software will be used, a solution was developed to deal with the limitation imposed by OPC DA, consisting in the deployment of an OPC DA to OPC UA wrapper software as interface to Linux software (*MatrikonOPC UA Wrapper for COM OPC Servers*). At the time of writing this document, the *Bosch-Rexroth* PLC firmware was upgraded to provide an OPC UA server that was under evaluation. The next step will be to create an ACS component for the drive system which will use the OPC Java DevIO class mentioned before.

After the ACS component has been tested and deployed, the drive control software will be used to interface the drive test stand. When the prototype is deployed a first phase of commissioning will take place, when the control software will be used to tune the system by using fiducial marks. In a later phase, the control system will be used to track astronomical objects and to check the pointing with the CCD cameras.

The CCD Cameras

The most important tool to check different aspects of the MST prototype will be a set of four CCD cameras. One goal of these CCD cameras will be to test the structural design of the prototype, and, with the help of a WS, the effect of the temperature, wind and other environmental factors. Additionally, the CCDs will be used to perform mirror adjustments, and will allow for measurements of the optical point spread function. The output images will be stored in the standard astronomical FITS data format. Excluding the emulated data sources, the CCD cameras will be the source of the largest fraction of data volume of the prototype.

The chosen CCD camera model 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 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. It is accessible by most programming languages such as C++ and Java (via JNI). The evaluation of this software has been positive, working correctly in the test machines in both implementation languages. The next step involves the implementation of an ACS component to operate the CCDs.

The Active Mirror Control

The design of the CTA telescopes makes use of tesselated reflector composed of individual mirror facets (see [8] for details). Each individual mirror facet will be attached to an AMC unit which has the functionality of allowing a perfect mirror alignment. This will allow online re-alignments of the LSTs reflectors, where the deformation caused by the weigh of the telescope will cause misalignments depending on the telescope elevation. It will also allow, for larger timescales, to perform re-alignments in MSTs and SSTs. The dish of the MST prototype will be completely covered by a mixture of real and dummy mirrors. Several AMC units of two different AMC design concepts will be installed in the prototype (see [1] 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 a PC via USB or RS-232 serial interface. The other type of unit is interfaced via CAN-Bus, accessed via a Ethernet-CAN-Bus gateway.

From the MST prototype control design point of view the idea is to unify the higher level interfacing to both unit types. This can be achieved with a higher lever ACS com-

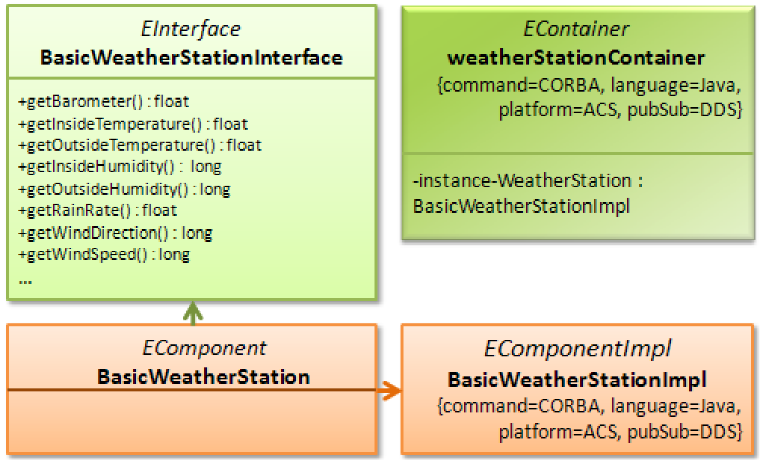

Figure 3: UML class diagram of an ACS component for the weather station for the prototype, modeled with MagicDraw.

ponent which sends the common instructions for the alignment procedures, acting as client for the lower level ACS components that will be different for each of the two AMC concepts.

For the AMC units using the XBee communication a preliminary version of a Python ACS component which uses the *Pyserial* library has been developed. Real movement of the AMC actuators was tested by using the ACS *Object Explorer* generic client. The movement of an AMC unit installed in the "quarter-dish" has also been recently tested.

The Weather Station

A WS has been acquired and will be installed nearby the MST prototype to continuously monitor the weather parameters, allowing correlation with the behavior of the structure of the MST with changes in the environmental status like, for example, the wind speed or temperature.

The chosen WS model is a *Davis Vantage VUE*. This WS is able to measure the wind speed and direction, as well as other quantities with the required accuracy and measurement rate. The instrument is composed of an outdoor unit communicating via a WPAN with an indoor unit, including a data-logger with some limited internal storage capacity and equipped with a RS-232 serial interface.

The WS is the least complex component of the MST prototype and for that reason has been the first device used to experiment with previously mentioned software development procedures. Starting from the lower level, an OPC UA server has been implemented with the OPC UA Java SDK which uses the RXTX library to communicate with the WS *datalogger* via the serial line. In a higher lever, a Java ACS component has been implemented using the UML-model based code generation framework (see Fig. 3), and tested with the ACS *object explorer*.

CONCLUSIONS

The MST prototype will act as the first realistic *test bench* for the control software for the CTA array. Equipped with complex systems like the drive system and AMC, the MST prototype will allow the evaluation of the ACS control middleware while the hardware design concepts are being tested, providing input and expertise to the software control developers of CTA. The use of tools like the UML code generation and the OPC UA servers, already tested the relatively easy system of the WS, will eventually allow the setting of a efficient framework for the CTA array control while optimize the productivity.

ACKNOWLEDGMENTS

We thank the excellent support of ACS experts from ESO Garching, in particular G. Chiozzi and H. Sommer. We want to acknowledge the help of A. Lopatin by teaching us ACS. We gratefully acknowledge support from the agencies and organisations listed in this page: http://www.ctaobservatory.org/?q=node/22

REFERENCES

[1] CTA Consortium, "Design Concepts for the Cherenkov Telescope Array", `arXiv:1008.3703`, 2010.

[2] E. Davis, et al. "Mechanical Design of a Medium-Size Telescope for CTA". CTA internal note.

[3] G. Chiozzi, et al., "The ALMA Common software: a developer friendly CORBA based framework", SPIE Procs. 5496-23 (2004).

[4] Le Flour, J.L. Panazol (LAPP/IN2P3). "OPCUA for CTA readout", CTA internal communication, Heidelberg Feb. 2011

[5] http://www.mpi-hd.mpg.de/hfm/HESS/

[6] Di Marcantonio, P., et al. "Evaluation of Software and Electronics Technologies for the Control of the E-ELT Instruments: a Case Study", in these proceedings.

[7] N. Troncoso, et al. "A Code Generation Framework for the ALMA Common Software", SPIE Procs. 7740-121 (2010).

[8] A. Föster, et al. "Mirror Development for CTA". Proc of the 32nd International Cosmic Ray Conference, Beijing, China, 2011.