PROCEEDINGS OF SCIENCE

ONLINE ICRC 2021
THE ASTROPARTICLE PHYSICS CONFERENCE
Berlin | Germany
37th International Cosmic Ray Conference
12–23 July 2021

The Architecture of ASTRI Mini Array Cherenkov Camera Software Supervisor

M. Corpora[a,*], A. Grillo[b], P. Sangiorgi[a], M. Capalbi[a], O. Catalano[a], G. Sottile[a], F. Russo[c], G. Tosti[g], A. Bulgarelli[c], F. Lucarelli[e], N. Parmiggiani[c], J. H. Schwarz[d], F. Russo[a] and S. Scuderi[f] for the ASTRI Project

[a] *INAF/IASF Palermo - Via Ugo La Malfa 153, Palermo, Italy*
[b] *INAF/OA Catania - Via S. Sofia 78, Catania, Italy*
[c] *INAF/OAS Bologna - Via P. Gobetti 93/3, Bologna, Italy*
[d] *INAF/OA Brera - Via Brera 28, Milano, Italy*
[e] *INAF/OA Roma - Via Frascati 33, Monte Porzio Catone, RM, Italy*
[f] *INAF/IASF Milano - Via Alfonso Corti 12, Milano, Italy*
[g] *Dipartimento di Fisica, Università degli Studi di Perugia, I-06123 Perugia, Italy*

E-mail: mattia.corpora@inaf.it, alessandro.grillo@inaf.it

The ASTRI Mini-Array (MA) is an INAF project to construct and operate an experiment for the study of gamma-ray sources emitting up to the TeV spectral band. The ASTRI MA consists of nine Imaging Atmospheric Cherenkov Telescopes that will be deployed at the Observatorio del Teide (Tenerife, Spain). These telescopes will be an evolution of the two-mirror ASTRI-Horn telescope, successfully tested since 2014 at the Serra La Nave Astronomical Station of the INAF Observatory of Catania. Each telescope will be equipped with the new version of the ASTRI Silicon Photo-Multiplier (SiPM) Cherenkov Camera. The ASTRI-MA will be monitored and controlled by a Supervisory Control And Data Acquisition (SCADA) system which consists of different software subsystems. One of these is the Cherenkov Camera Supervisor (CCS) that controls each Cherenkov Camera. Its main functionality is to realize an interface between each Camera and the central SCADA software. The CCS provides the services to control and monitor the Camera through the Alma Common Software (ACS). This is a framework based on object-oriented CORBA middleware, which gives the infrastructure for the exchange of messages between distributed objects and system-wide services. The CCS is based on the Open Platform Communications - Unified Architecture (OPC-UA) protocol, a platform-independent service-oriented architecture. This work presents the design and the technologies used by the ASTRI Camera team to implement the CCS. It describes the Mini Array Software System Structure and CCS architecture and functionalities.

37th International Cosmic Ray Conference (ICRC 2021)
July 12th – 23rd, 2021
Online – Berlin, Germany

[*] Presenter

1. Introduction

The Italian National Institute for Astrophysics (INAF) is leading the ASTRI Mini-Array (MA) project. It concerns the construction of nine Cherenkov Telescopes at the Observatorio del Teide (Tenerife, Spain) in order to study gamma-ray sources, emitting at very high energy in the TeV spectral band, and perform intensity interferometry observations. These telescopes will be an evolution of the two-mirror ASTRI-Horn telescope[1] [2], installed in 2014 at the Serra La Nave Astronomical Station of the INAF Observatory of Catania (Italy), and successfully tested with the detection of the Crab Nebula [3]. ASTRI-Horn included the development of a Cherenkov Camera [4] that consists of various hardware subsystems controlled by an electronics central unit, the Back-End Electronics (BEE) [5] and a Camera Control Software [6].

ASTRI MA is a very complex system made up of several interconnected hardware and software subsystems. For this reason, a Supervisory Control And Data Acquisition (SCADA) system is needed. It is a distributed software system that supervises, controls and acquires the data from the Mini-Array system to perform scientific observations. The Operator Human Machine Interface (HMI) is part of the SCADA system. SCADA has a Central Control System which interfaces and communicates with all ASTRI MA subsystems and their dedicated software installed at the site. SCADA is responsible for the execution of the observations and shall normally perform the operations in an automated way but is supervised by the Operator located in one of the ASTRI Control rooms. Furthermore, SCADA shall collect scientific data provided by the scientific instruments, logging data, monitoring data, alarm data provided by the ASTRI MA subsystems and shall provide online observation quality information to the Operator in order to assess the quality of data during the acquisition. The SCADA/Central Control System, including the control software subsystems and low-level software components, shall be built atop the same framework that has been successfully used by ALMA, namely, the ALMA Common Software (ACS) [7]. ACS is a container/component framework, designed for distributed systems, with standardized paradigms for logging, alarms, location transparency, and supports three programming languages: Java, C++ and Python. Among the software subsystems, the *Telescope Control System* (TCS) coordinates all telescope assemblies. It shall use the services offered by *Supervisors*, software components that interact with the Local Control Software of the assemblies. One of the Supervisors is the Cherenkov Camera Supervisor (CCS), which provides the services for the control of the Cherenkov Cameras. Its main functionality is to realize an interface between each Camera and the TCS manager component, called *Telescope Manager*. The CCS provides the services to control and monitor the Camera through the ACS. The following sections describe the CCS software architecture, the provided and requested interfaces and the interactions with the other software components. Also, we describe the context of the system of which the CCS is a part.

2. Astri MA Cherenkov Camera

The Cherenkov Cameras shall be composed of four hardware subsystems: *Mechanical* subsystem, *Power Supply* subsystem, *Electronics* subsystem and *Thermal System*. The Mechanical subsystem has the main function of containing the entire camera electronics, including power distribution system, cooling/heating thermoelectric system, fiber-optic calibration system and ancillary devices. It provides the mechanical interface to the telescope structure. The lid and the Photo Detection Modules (PDMs) support structure are also included. The Power Supply subsystem has the main function of generating and distributing the required voltages among the various electronics components. The Electronics Assembly comprises Photo Detection Modules (PDMs), Back-End Electronics (BEE) and the ancillary electronics. The PDMs include SiPMs, Front-End Electronics (FEE) and Field Programmable Gate Array (FPGA) Printed Circuit Boards (PCBs). The FEE contains Application Specific Integrated Circuits (ASICs) that directly interface the SiPMs. The main function of FFE is to convert the analog SiPM signals in digital signals. The FEE also provides discriminator outputs that are used to produce a local trigger. The FPGA PCBs implement the algorithm for the generation of the local trigger, the logic for event acquisition and the transmission of the data produced to the BEE. The BEE controls and manages the overall system, including data management formats, Lid mechanisms and fiber-optic calibration tool.

3. Mini-Array Software System Structure

To describe the CCS architecture, it is useful to explain some definitions and the software organization. All components described below, include a hardware part and related software. A *system* is an arrangement of parts that together exhibit behaviour or meaning that the individual constituents do not. A *subsystem* is a system in its own right, except it normally will not provide a useful function on its own, it must be integrated with other subsystems (or systems) to make a system. The term *assembly* is used to indicate a hardware and software part of a subsystem. The term *device* is used to indicate a part of an assembly.

The SCADA system shall switch-on, switch-off, control, configure, get status and manage the subsystem state machine and assembly state machine, acquire monitoring points that can generate warnings or critical event notifications, errors and alarms of the assemblies of the ASTRI Mini-Array Observing Subsystems. Among the subsystem of SCADA, there is the Telescope Control System (TCS), which manages a single Telescope. The TCS shall be responsible for coordinating all telescope assemblies, starting up, configuring and shutting down the assemblies of the Telescope, supervising optical system control, telescope mount control and instrument control (Cherenkov Camera, Optical Camera). The main components of the TCS shall be:

- Cherenkov Camera Supervisor: the software component that controls, monitors and provides an interface to the Camera Local Control Sofwtare (LCS).
- SI3 Supervisor: the software component that provides an interface to the SI3 LCS;
- Mount Supervisor: the software component that controls and monitors the mount LCS and other auxiliaries;
- Optics Supervisor: the software component that controls and monitors the Optics LCS and the Optical Camera LCS;
- Telescope Manager (TM): the software component responsible for coordinating all TCS subsystems and starting up and shutting down the system.

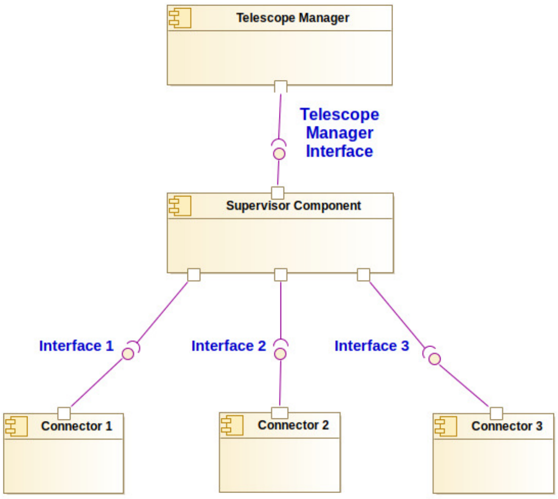

Figure 1: UML Component Diagram of Telescope Control System

All Supervisor components interface with Connectors and Telescope Manager. A Connector is a low-level software component whose function is to communicate with LCS via OPC-UA [8] to control devices. On the other hand, the TM is a software component that uses the business logic of the various supervisors and offers high-level services to the Central Control System. The Fig.1 shows a Unified Modeling Language (UML) generic component diagram which contains a supervisor, some connectors, the telescope manager and the relative interfaces.

The figure shows that a single assembly can be controlled by multiple Connectors for the purposes of modularization. The Cherenkov Camera, for example, consists of several devices which are managed separately by the back-end electronics.

4. Cherenkov Camera Supervisor Architecture

The CCS has two main features:

- combines the services offered by Camera Connectors in order to perform high-level operations;
- provides a high-level interface to Telescope Manager.

Fig.2 shows a context component diagram of the CCS. It interfaces with the following Connectors:

- Timing Connector: a component to manage the time synchronization device;
- VDB Connector: a component to monitor and control the voltage distribution device;
- LID Connector: a component to monitor and control the LID motion;
- TEC Connector: a component to monitor and control the thermal system;
- Focal Plane Connector: a component to manage the Camera Focal Plane;

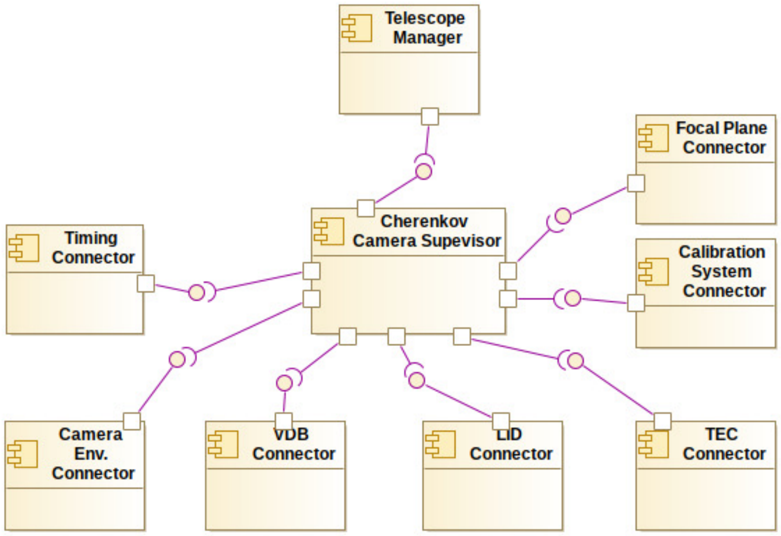

Figure 2: Cherenkov Camera Supervisor UML Context Component Diagram

- Calibration System Connector: a component to manage the Calibration System;
- Camera Environment Connector: a component to monitor the Camera environmental parameters, such as temperature and humidity.

The Connectors offer services for accessing OPC-UA nodes such as read/write an OPC-UA variable or call an OPC-UA method. They also provide a range based alarm system, ensuring that the values of the variables remain within the predetermined thresholds.

The CCS offers some high-level services to Telescope Manager through an interface. The main high-level services are: *Put online the Camera LCS*, *Stop the Camera LCS*, *Perform Camera Calibration* and *Perform a Cherenkov Observation* These services consist of the composition of primitive services offered by the Connectors. We describe some of the services in terms of operations performed by the CCS. To perform *Put on line the Camera LCS*, the CCS

- commands the Camera LCS, through the TEC Connecotor, to turn on the active control of the Thermal Control System;
- commands the Camera LCS, through the VDB Connector, to turn on the low and high voltages of the active PDMs of the focal plane;
- commands the Camera LCS, through the Focal Plane Connector, to initialize the focal plane with the default configuration;
- commands the Camera LCS to start the acquisition of HouseKeeping and Variance data.

In order to *Perform Camera Calibration*, the CCS

- commands the Camera LCS to turn on and configure the calibration system through the Calibration System Connector;

- configures for the data taking all the enable PDMs of the focal plane;
- starts acquisition and commands the Camera LCS to enable the data transmission;
- stop the acquisition and turns off the calibration system through the relative Connector.

In order to *Perform a Chrenkov Observation*, the CCS

- commands the Camera LCS to open the lid;
- through the Focal Plane Connector, configures all the enabled PDMs with the parameters in according to the scheduled observation (thresholds and trigger configurations);
- starts acquisition and enables the data transmission of the packets;

The CCS also manages the state machine (that is described below) of the Cherenkov Camera. Transitions from one state to another can be performed at the request of the TM or when errors and exceptions occur during the CCS operations.

4.1 CCS implementation

The CCS shall be developed in Java Programming Language. It consists of some classes interacting with each other. In particular, a CCS main class instantiates objects of the controller classes. Each controller class manages a single device through the relative Connector. Each Connector is created using ACS starting from a document that describes its functions, the so-called IDL (Interface Description Language). Through this document, the component skeleton is generated, which is then completed with the implementation part, i.e. the part of code that carries out the communication with the various nodes of the OPC-UA server. Fig. 3 shows the CCS class diagram. The *CameraSupervisorImpl* class implements the *CameraSupervisorInterface*, which contains the high-level operations (such as *performAcquisition*, *performCalibration* or *initCherenkovCamera*). The Telescope Manager shall use these ones according to the requests of the Central Control System. Each high-level operation corresponds to a sequence of calls to operations offered by the controllers. The latter will use the connectors services to communicate with the OPC-UA server, which will interact with the Cherenkov Camera LCS. In order to provide its state to Telescope Manager, Cherenkov Camera Supervisor also must implement a common assembly state machine shown in Fig. 4. Camera states are defined as follows:

- Off State: the Camera is entirely without electrical power.
- On State: with the following sub-states:
  - Initialised State: the state of the Camera after power on.
  - Standby State: a state that is still safe with respect to extreme conditions, but has all components activated, with preparations for observation initiated.
  - Operational State: a state associated with operations (e.g. data taking), with configuration dictated by performance requirements. Two sub-states could be present:
    - Nominal: the Camera can be operated with full performance.

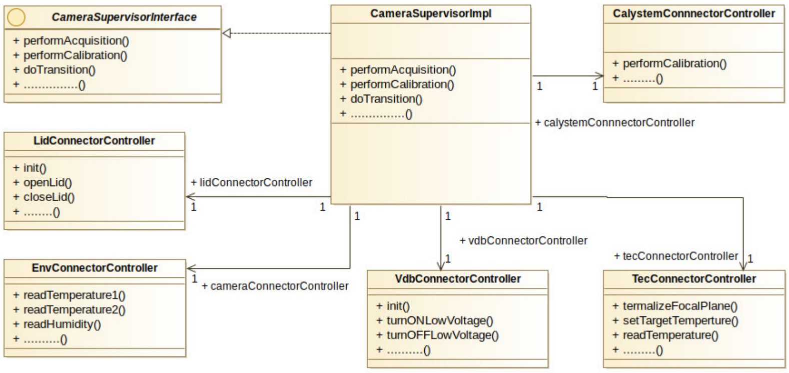

Figure 3: Cherenkov Camera Supervisor UML Class Diagram

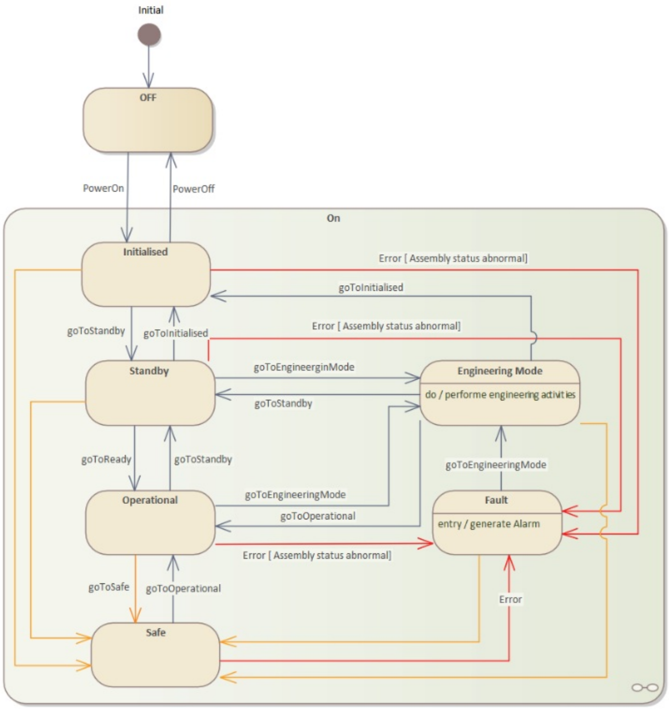

Figure 4: Cherenkov Camera Supervisor State Machine

- ∗ Degraded: the Camera can be operated with reduced performance.

- Safe State: if dangerous conditions are present, the Camera goes into a configuration where the object is considered exposed to "normal" risk for damage or loss.
- Fault State: the Camera has encountered a serious problem, which means it is currently unable to meet the requirements associated with one of the standard states.
- Engineering Mode State: a state designed to facilitate maintenance and engineering activities, and is unavailable for routine operations.

The *CameraSupervisorImpl* class shall also provide some services regarding logging and alarm management to the Telescope Manager. In particular, it must be able to raise alarms and warnings whenever a device or its relative controller faults.

5. Conclusion

This design will be the basis for the software development of the Cherenkov Camera Supervisor in ASTRI Mini Array software. Even if the architecture is designed independently of the hardware that will be used, it could be changed and updated when the first prototype of the Cherenkov Camera will be built. As the system was designed, it respects software engineering criteria, such as modularization, granularity and object orientation. This could favor future hardware and software integration, as well as changes to existing modules.

Acknowledgments

This work was conducted in the context of the ASTRI Project. This work is supported by the Italian Ministry of Education, University, and Research (MIUR) with funds specifically assigned to the Italian National Institute of Astrophysics (INAF). We acknowledge support from the Brazilian Funding Agency FAPESP (Grant 2013/10559-5) and from the South African Department of Science and Technology through Funding Agreement 0227/2014 for the South African Gamma-Ray Astronomy Programme. IAC is supported by the Spanish Ministry of Science and Innovation (MICIU).

References

[1] G. Pareschi, *The ASTRI SST-2M prototype and mini-array for the Cherenkov Telescope Array (CTA)*, Proc. SPIE 9906 in proceedings of *Ground-based and Airborne Telescopes VI*, (2016).

[2] M. C. Maccarone, *ASTRI for the Cherenkov Telescope Array*, in proceedings of *Proceedings of 35th International Cosmic Ray Conference*, 2017.

[3] S. Lombardi et al., *First detection of the Crab Nebula at TeV energies with a Cherenkov telescope in a dual-mirror Schwarzschild-Couder configuration: the ASTRI-Horn telescope*, *A&A* (2020) 634 A22.

[4] O. Catalano et al., *The ASTRI camera for the Cherenkov Telescope Array*, in proceedings of *Ground-based and Airborne Instrumentation for Astronomy VII (SPIE)*, (2018).

[5] G. Sottile, *ASTRI SST-2M camera electronics*, in proceedings of *Ground-based and Airborne Telescopes VI (SPIE)*, (2016).

[6] P. Sangiorgi et al., *The ASTRI Camera control software of the ASTRI SST-2M prototype for the Cherenkov Telescope Array*, *Nuclear and Particle Physics Proceedings*(2019) 306-308.

[7] J. H. Schwarz et al., *The ALMA software architecture*, in proceedings of *Advanced Software, Control, and Communication Systems for Astronomy (SPIE 5496)*, (2004).

[8] OPC Unified Architecture, *Specification, Part 1: Overview and Concepts*, Release 1.02, July 10, 2012."