Publication Year	2022
Acceptance in OA	2025-01-21T11:17:12Z
Title	The telescope control system for the ASTRI Mini-Array of imaging atmospheric Cherenkov telescopes
Authors	RUSSO, Federico, TOSTI, Gino, BRUNO, Pietro Giuseppe, BULGARELLI, ANDREA, GIANOTTI, Fulvio, PARMIGGIANI, Nicolo', CAPALBI, Milvia, CONFORTI, Vito, CORPORA, Mattia, COSTA, Alessandro, FIORI, Michele, GERMANI, STEFANO, GRILLO, Alessandro, SANGIORGI, Pierluca, SCHWARZ, Joseph Hilary, SCUDERI, Salvatore, ZAMPIERI, Luca, INCARDONA, Federico, MUNARI, Kevin, PASTORE, Valerio
Publisher's version (DOI)	10.1117/12.2629943
Handle	http://hdl.handle.net/20.500.12386/35671
Serie	PROCEEDINGS OF SPIE
Volume	12189

The telescope control system for the ASTRI Mini-Array of imaging atmospheric cherenkov telescopes

Federico Russo[*a], Gino Tosti[b], Pietro Bruno[c], Andrea Bulgarelli[a], Fulvio Gianotti[a], Nicoló Parmiggiani[a], Milvia Capalbi[d], Vito Conforti[a], Mattia Corpora[d], Alessandro Costa[c], Michele Fiori[e], Stefano Germani[b], Alessandro Grillo[c], Pierluca Sangiorgi[d], Joseph Schwarz[f], Salvatore Scuderi[g], Luca Zampieri[e], Federico Incardona[c], Kevin Munari[c], Valerio Pastore[a] for the ASTRI Project[**];
[a]INAF - Osservatorio di Astrofisica e Scienza dello Spazio, Via Gobetti 93/3, 40129, Bologna, Italy;
[b]Universitá degli studi di Perugia, Piazza Università, 1, 06123, Perugia, Italy;
[c]INAF - Osservatorio Astrofisico di Catania, Via S. Sofia 78, 95123, Catania, Italy;
[d]INAF - Istituto di Astrofisica Spaziale e Fisica Cosmica, Via Ugo la Malfa 153, 40196 Palermo;
[e]Universitá degli studi di Padova, dipartimento di Fisica e Astronomia, Via Francesco Marzolo, 8, 35121, Padova, Italy;
[f]INAF - Osservatorio Astronomico di Brera, Via Brera, 28, 20121, Milano, Italy;
[g]INAF - Istituto di Astrofisica Spaziale e Fisica cosmica, Via Alfonso Corti, 12, 20133, Milano, Italy;

ABSTRACT

The ASTRI Mini-Array is an international project led by INAF to construct and operate nine Imaging Atmospheric Cherenkov Telescopes with the scientific goals of studying several classes of objects possibly emitting at energies higher than some TeVs and of performing stellar intensity interferometry observations. The telescopes array will be installed at the Teide Observatory (Tenerife, Spain). A Supervisory Control And Data Acquisition (SCADA) software system will be developed to manage the ASTRI Mini-Array allowing its control remotely, from several locations. One of the most important components of the SCADA system is the Telescope Control System (TCS), i.e. the system responsible for the control and supervision of each telescope. The TCS includes several supervisor components, that interface with the telescope local control systems, the hardware and software that control the telescopes hardware devices such as the telescope mount drive systems and the Cherenkov camera, via the Open Platform Communications - Unified Architecture (OPC-UA) standard. These supervisors are then controlled by a telescope manager component responsible for the execution of the single telescope scientific and technical operations requested, orchestrated and synchronized centrally by the SCADA array central controller. This contribution describes the TCS architecture, design and development approach in the context of the general SCADA architecture and of the ALMA Common Software, the framework chosen for the development of all SCADA software of the ASTRI Mini-Array.

Keywords: Control software, ASTRI Mini-Array, Telescope, Cherenkov, Supervisory

1. INTRODUCTION

The ASTRI Mini-Array is an INAF project to construct, deploy and operate an array of nine Cherenkov 4-meter class telescopes at the Teide Observatory in the Canary Islands. The project is developed in collaboration with University of Sao Paulo - Brazil, North-West University - South Africa and Instituto de Astrofisica de Canarias (IAC) - Spain. The ASTRI Mini-Array technology is based on the ASTRI-Horn prototype, a small-sized Cherenkov telescope (SST) developed by INAF within the Cherenkov Telescope Array (CTA) Project and located in Italy at the INAF "M.C. Fracastoro" observing station (Mt. Etna, Sicily)[1]. The ASTRI Mini-Array is designed to be an end-to-end system compliant with strict requirements. For this reason, industrial standards, proven technologies and best practices have been

* federico.russo@inaf.it; phone +39 051 639 8665
** http://www.astri.inaf.it/en/library/

Software and Cyberinfrastructure for Astronomy VII, edited by Jorge Ibsen, Gianluca Chiozzi, Proc. of SPIE Vol. 12189, 121892I · © 2022 SPIE
0277-786X · doi: 10.1117/12.2629943

employed thanks to the ASTRI-Horn experience [2]. In the framework of the ASTRI Mini-Array, the task of the telescope control system (TCS) is to manage individually each telescope regarding supervision, control and monitoring of the telescope itself and its subsystems, as well as engineering operations.

2. DESIGN

The first point to be addressed for a control software is that of its architecture as it is made up of numerous specialized subsystems. That is, it must be able to orchestrate the flow of data between heterogeneous structures and computer programs, allowing them to communicate despite the diversity of protocols or operating systems used. In order to meet the functional and safety requirements, it is also necessary to use a highly reliable software system that provides for the use of widely used industrial standards in order to ensure its robustness. The main telescope control system components, which implement the required scenarios, are depicted in the gray rectangle of Fig.1, the components that are depicted outside this rectangle represent the software that TCS interface with.

2.1 Static view

Each Telescope subsystem (e.g. Mount, Optical Assembly, Cherenkov Camera, etc.) has its own **local control system** that interacts with the relevant hardware devices. The assemblies are grouped in a control hierarchy, in which each parent consists in a **supervisor** module that relays the commands to its children. The supervisors are the low level part of the TCS and manage the hardware by acting as a client to the local control software that executes their commands acting as server.

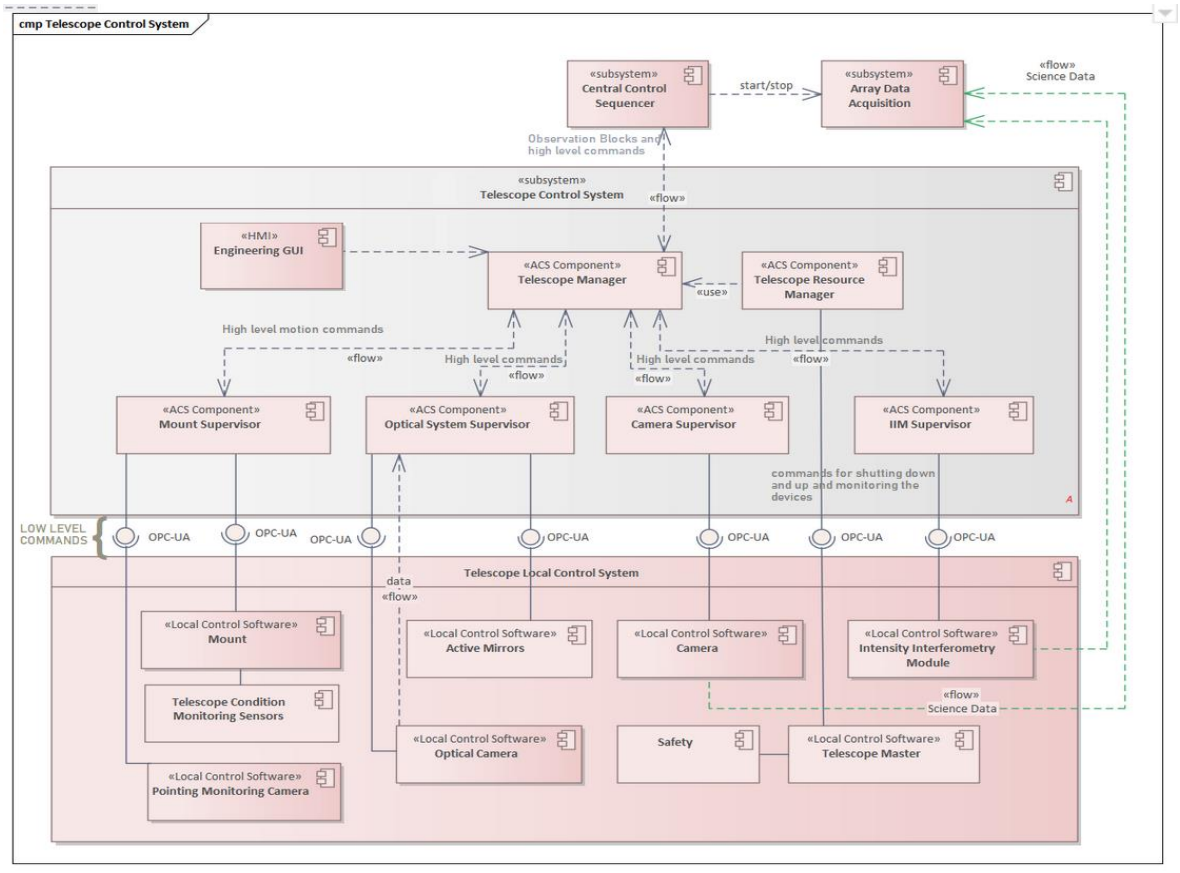

Figure 1. TCS main components and its context.

At the TCS core level the Telescope Manager module provides all the common services necessary for observations, logging and alarm management. Its main function is to translate a complex action, such as the tracking of a gamma source, into a logical balance of input and output, among the supervisors, that concretizes the tracking within the requirements. The

supervisors themselves manage actuators and sensors of the telescope by interacting with the local control software which represent the low level interfaces for the TCS. The most important supervisors are the Mount Supervisor (MS) and the Cherenkov Camera Supervisor (CCS). Both supervisor offer some high-level services to the Telescope Manager through an interface. The main high-level services for the CCS are: Put online the Camera LCS, Stop the Camera LCS, Perform Camera Calibration and Perform a Cherenkov Observation [5] while the main high-level services for the MS are goToAzEl, goToSkyTarget, trackSkyTarget, stopMotion, park and unpark the telescope. These services consist of the composition of primitive services offered by the connectors, whose components are described below in section 3.3. The Telescope Manager can receive, in a mutually exclusive way, high level commands from the engineering GUI, which is part of the TCS, and from the Central Control, which is external to the TCS and represents its higher interface in normal operation conditions (I.E. non-engineering). The Central Control is the software system that controls all the operations carried out at the ASTRI Mini-Array site. SCADA (Supervisory Control and Data Acquisition) has a Central Control System that interfaces and communicates with all assemblies and dedicated software installed at the site. It is responsible for the execution of the Scheduling Blocks to perform observations.

2.2 Dynamic view

The TCS architecture briefly described in the previous section is a static and high-level description of the system. To discuss in greater detail its most important components from a dynamical point of view, it is useful to analyze an excerpt from a use case.

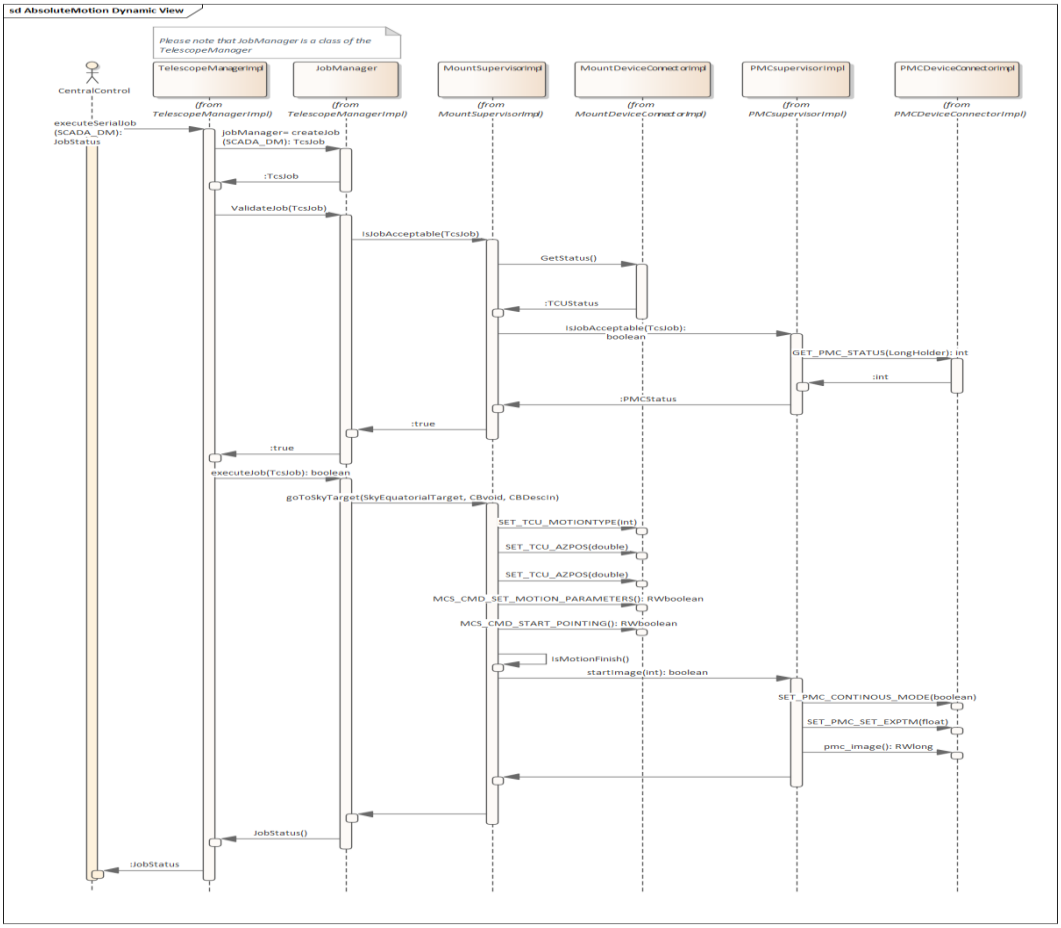

Figure 2. Pointing sequence diagram.

The use case briefly described in this section regards the Central Control which commands the TCS to execute a pointing in the sky: the command is sent directly to the Telescope Manager which initializes the coordination of all the subsystems that will take part in the workflow. As shown in Fig.2, the first task performed by Telescope Manager, is to command the

Job Manager to create the internal data type derived from the received command. This step allows to hide the external data definitions from the internal logic structure and implementation of the TCS. This means that only the Job Manager, that is a part of the Telescope Manager, must be aware of the translation from external into internal commands. Once the internal data structure, called TcsJob, is created the Telescope Manager asks the relevant supervisors, involved in the specific operation, to validate the TcsJob itself. This process consists in checking whether the requested operation can be executed within the current status of the system and/or the environment. Another check is done on the command data formal correctness. In order to finalize the validation, the supervisors use the device connectors to collect the status of the hardware and, once obtained, verify if the hardware is able to accept the relevant commands. Please refer to section 3.3 for a description of device connectors to the local control software. Now the TcsJob has been validated and the Telescope Manager begins to orchestrate the executions of the specific commands, in the exact required order, for the supervisors in charge of performing the pointing procedure. The mount supervisor interacts with the mount device connector by dispatching information about the sky target and issuing the commands to start the pointing. Finally, the accuracy of the pointing is monitored by means of the astrometry of the images taken by the PMC (Pointing Model Camera) which is activated and managed by the PMC supervisor.

3. DEVELOPMENT APPROACH

3.1 High level implementation

The TCS receives high level commands, and transmits monitoring information, both from/to the Central Control and the Engineering Graphical User Interface. As the Central Control is not part of the TCS its description is out of the scope of this paper and will not be further described (a brief description can be found in section 2.1). On the other hand, the Engineering GUI, represents the highest software level of the TCS and its role is to offer to the telescope engineering operator control commands and monitoring information on the system. The adopted GUI will be strongly derived from the ASTRI-Horn GUI [2] that is shown in Fig.3. The GUI is the point where monitoring tools, tools for accessing real time data, controls of high or low level operations on the telescope and on all subsystems, converge. The main client is not a software system, but the human operator. For this reason, the GUI incorporates concepts of Human Machine Iteration to make its use intuitive and fast, to prevent and correct unauthorized actions, highlight alarms that require prompt action by the operator, etc. The TCS GUI is implemented in JavaFx which is a family of application software, based on the Java platform, for the creation of rich Internet applications, i.e. web applications that possess the characteristics and functionalities of desktop applications, without requiring installation on the hard disk. The GUI would allow the user to access the control software, and its various tools, even remotely. Its architecture implements:

• a set of semaphores to coordinate access to shared resources in the case of multiple simultaneous remote accesses. For example, only one operator at a time can access the movement of the telescope, but multiple operators can monitor the data.

• a stateless philosophy to unlink the behavior of the GUI to any status of the telescope. This is an important feature because at every restart of the GUI, caused by any reason, such as a power failure or an error, the GUI must return to the identical previous state based only on the output of the telescope system and not on the state of the GUI itself. This always ensures consistency between the GUI and the control system.

• a software system based on multi-threading to optimize performance. To obtain a stateless GUI, multi-threading must be based on continuous cycles, independent on each other and therefore asynchronous. Each thread checks the telescope subsystems providing monitoring values, graphs, and statistical values.

The Java platform has been chosen as it provides a complete set of concurrent and distributed development libraries available through the java.util.concurrent package. The javafx.concurrent package includes the Worker interface and two concrete implementations, Task and Service classes. The Worker interface provides APIs to allow a background thread to communicate with the user interface. The Task class allows developers to implement asynchronous tasks in JavaFX applications.

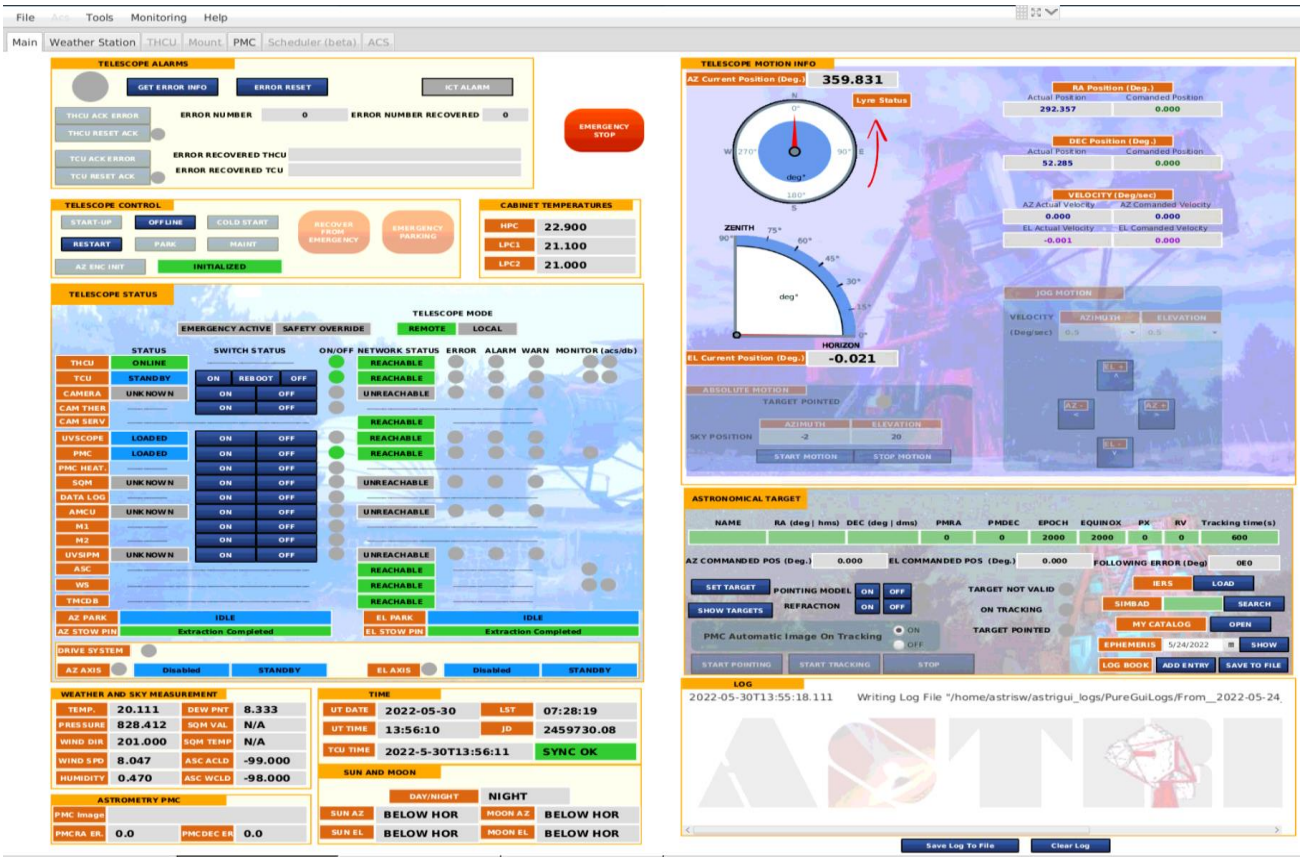

Figure 3. ASTRI-Horn engineering GUI.

3.2 Middle level implementation

The GUI described above does not have direct access to the hardware, but is connected to it via a middle level software. That is the central part, at a logical level, for the TCS. It concretizes the concept of interoperability, i.e. the ability of a system to interact and function with other products or subsystems, existing or still in progress, with no restrictions on access or implementations. In order to achieve this concept, two implementation issues need to be addressed by the TCS:

• A transmission protocol that acts as a common language between the subsystems.

• A middleware capable of coherently orchestrating the data flows between the subsystems.

An optimal solution adopted for the TCS, in order to fulfill the first need is the Open Platform Communications' Unified Architecture (OPC-UA), an automatic communication protocol for industrial automation. Produced by the OPC foundation, it is suitable for multiple platforms, open and secure, allowing programmable logic controllers (PLCs), both new and previous generation, to communicate. Both Java and the low-level software system that we will discuss later are capable of using this protocol. The OPC-UA is therefore the glue that allows the parts of the system to be held together by defining interfaces through OPC-UA variables, but the data flow must still respect specific patterns: we then come to the other implementation problem. The solution implemented to resolve the middleware problem is the use of the Alma Common Software (ACS) framework which provides a common infrastructure for all orchestrated subsystems. The heart of ACS is based on a model of distributed components implemented as CORBA objects [3]. CORBA (Common Object Request Broker Architecture) is a standard developed by the object management group (OMG), a consortium created in 1989 with 440 companies including: Microsoft and Sun to allow communication between components regardless of their distribution on the different nodes of the network or the programming language with which they were developed. ACS is successfully used by the team responsible for developing the Atacama Large Millimeter Array (ALMA) control system which uses ACS components for high-level entity control. Similarly, it will be used for the CTA project. ACS offers

common services such as communication between components, management of their life cycle, propagation of errors and alarms, interaction with the database, etc.

3.3 Low level implementation

The very low level for the TCS consists in a set of software components called device connectors. The device connectors components are implemented upon the same software frameworks chosen for the middleware: the ACS middleware, which provides common ways to access the hardware, such as OPC-UA protocol, together with monitoring, alarm, and logging services support. Thus, the use of OPC-UA allows the decoupling of the access peculiarities of each assembly with the hardware control systems. The key concept is that each device connector is used by the relevant supervisor in order to supervise, control and monitor the local control software and so the telescope hardware itself. Each interface with a hardware component accessed via OPC-UA is described by an Interface Control Document (ICD). In ASTRI these ICDs take the form of tables in which each row is a command of four different types: monitoring points (GET), configuration points (SET), functional execution commands (CMD) and state transition commands (MODE). For each control or monitoring point a complete description of the information required, e.g. data type, OPC-UA node, connected alarms, is provided. We use the ICD format not only to document the interface, but also to help the developers in the task of producing all the support code that depends on non-device-specific logic, via a code generator. The ASTRI code generator is implemented in a few Python scripts and the output is depicted in Fig. 4 [4]. Each ACS device connector component is thus generated for hardware ICDs with a set of templates to control the assemblies together with simple simulators at the OPC-UA level and engineering user interfaces to test them.

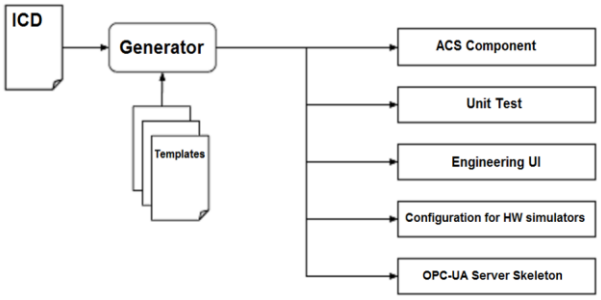

Figure 4. Code Generation workflow.

Although the first software layer upon the hardware, that is composed by the local control software, is not part of the TCS itself, it is useful to mention this level as it represents an extremely important context that has influenced the choices of the implementation of the TCS itself. Within this level exist more stringent requirements regarding both performance, reliability and safety, since this is where the real handling logic resides as well as the safety system of the telescope. Delegating the security protocol to a higher level would not be conceptually correct since its operation would result from the sum and interaction of several algorithms with the negative effect of introducing unsystematic potential bugs and therefore difficult to detect during testing and debugging. ACS is conceived as a framework for heterogeneous subsystems, while at this level, given the very close interaction with the hardware, it is necessary that the control of the data flow is carried out by a highly specialized middleware. For this reason, the TCS only checks the formal correctness of the commands received before forwarding them to the low level. For example, if the telescope cannot be moved because the cabinet door is open, the low level will notice this and refuse to operate a telescope motion by signaling the alarm to the TCS. This is the reason why these kinds of controls are done by an external (to TCS) Safety system which is highly interconnected with the hardware and is capable of blocking or inhibiting the handling hardware even bypassing the control software itself. This is a fundamental safety requirement in the event of an emergency (e.g. reaching travel limits) and for the risk of personal accidents (e.g. accidental handling of the telescope). It has been chosen to delegate the telescope handling calculations to this level as well to ensure that the aiming and tracking meet the accuracy requirements. In particular, the interaction between the encoders and the actuators must have a frequency of the order of milliseconds. Furthermore, programming numerically controlled systems provide handling libraries and PID (or PI) controls on the actuators. PID stands for Proportional-Integral-Derivative (control) and is a negative feedback system widely used in

control systems. These are algorithms that compute the inputs which determine the current value, and react to any positive or negative error. Regarding the telescope motion the TCS interacts, via OPC-UA, with Beckhoff TwinCAT software system that resides in this level. TwinCAT (The Windows Control and Automation Technology) is a platform that meets all the above requirements by transforming a windows PC into a real-time controller with a multi-PLC system, NC axis control, PID controls, programming environment, station operational and integration of the Safety system. The internal communication, between the hardware devices of the mounting system, uses the Ethernet Control Automation Technology (EtherCAT) protocol, ensuring reliability and performance in real time for the development of control and safety applications.

4. CONCLUSIONS

In the ASTRI Mini-Array approach the telescope control system is the principal software component in charge of the coordination and management of each of the nine telescopes, that will be installed to the Teide Observatory, for scientific and engineering functionalities, while its engineering GUI is the software module dedicated to the user operation for engineering operations. In this paper we showed a brief view of the architecture of the TCS focusing on its high, middle and low software levels.

5. ACKNOWLEDGEMENTS

This work was conducted in the context of the ASTRI Project thanks to the support of the Italian Ministry of University and Research (MUR) as well as the Ministry for Economic Development (MISE) 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). This work has also been partially supported by H2020-ASTERICS, a project funded by the European Commission Framework Programme Horizon 2020 Research and Innovation action under grant agreement n. 653477. The ASTRI project is becoming a reality thanks to Giovanni “Nanni” Bignami, Nicolò “Nichi” D’Amico two outstanding scientists who, in their capability of INAF Presidents, provided continuous support and invaluable guidance. While Nanni was instrumental to start the ASTRI telescope, Nichi transformed it into the Mini Array in Tenerife. Now the project is being built owing to the unfaltering support of Marco Tavani, the current INAF President. Paolo Vettolani and Filippo Zerbi, the past and current INAF Science Directors, as well as Massimo Cappi, the Coordinator of the High Energy branch of INAF, have been also very supportive to our work. We are very grateful to all of them. Nanni and Nichi, unfortunately, passed away but their vision is still guiding us. This article has gone through the internal ASTRI review process.

REFERENCES

[1] L.A. Antonelli on behalf of the ASTRI Collaboration “The ASTRI Mini-Array at Teide Observatory” Proceeding of Science, 2022, https://doi.org/10.22323/1.395.0897

[2] Tanci, C., Tosti, G., Conforti, V. “The ASTRI Mini-Array software system (MASS) implementation: a proposal for the Cherenkov Telescope Array” Proc. SPIE 2016, https://doi.org/10.1117/12.2231294

[3] G. Chiozzi et al., "The ALMA common software: a developer friendly CORBA-based framework" in Proc. SPIE Astronomical Telescopes + Instrumentation, Kona, United States, July 2004.

[4] Antolini, E., Tosti, G., Canestrari, R. “Telescope control system of the ASTRI SST-2M prototype for the Cherenkov Telescope Array”, Proceedings of ICALEPCS2017, doi:10.18429/JACoW-ICALEPCS2017-THMPL

[5] Corpora, M., Grillo, A., Sangiorgi, P. “Design and development of the Supervisor Software Component for the ASTRI Mini-Array Cherenkov Camera” Proc. SPIE 2022 - Paper 12189-72