JATIS Journal of Astronomical Telescopes, Instruments, and Systems

RESEARCH PAPER

Software architecture and development approach for the ASTRI Mini-Array project at the Teide Observatory

Andrea Bulgarelli,[a,*] Fabrizio Lucarelli,[b,c] Gino Tosti,[d] Vito Conforti,[a]
Nicolò Parmiggiani,[a] Joseph Hillary Schwarz,[e,†]
Juan Guillermo Alvarez Gallardo,[f] Lucio Angelo Antonelli,[b] Mauricio Araya,[f]
Matteo Balbo,[g] Leonardo Baroncelli,[a] Ciro Bigongiari,[b] Pietro Bruno,[h]
Milvia Capalbi,[i] Martina Cardillo,[j] Guillermo Andres Rodriguez Castillo,[i]
Osvaldo Catalano,[i] Antonio Alessio Compagnino,[i] Mattia Corpora,[i]
Alessandro Costa,[h] Silvia Crestan,[k] Giuseppe Cusumano,[i] Antonino D'Aì,[i]
Valentina Fioretti,[a] Stefano Gallozzi,[b] Stefano Germani,[e] Fulvio Gianotti,[a]
Valentina Giordano,[h] Andrea Giuliani,[k] Alessandro Grillo,[h] Isaias Huerta,[f]
Federico Incardona,[h] Simone Iovenitti,[e] Nicola La Palombara,[k]
Valentina La Parola,[i] Marco Landoni,[e] Saverio Lombardi,[b]
Maria Cettina Maccarone,[i] Rachele Millul,[e] Teresa Mineo,[i] Gabriela Montenegro,[f]
Davide Mollica,[i] Kevin Munari,[h] Antonio Pagliaro,[i] Giovanni Pareschi,[e]
Valerio Pastore,[a] Matteo Perri,[b] Fabio Pintore,[i] Patrizia Romano,[e]
Federico Russo,[a] Ricardo Zanmar Sanchez,[h] Pierluca Sangiorgi,[i]
Francesco Gabriele Saturni,[b] Nestor Sayes,[f] Eva Sciacca,[h] Vitalii Sliusar,[g]
Salvatore Scuderi,[k] Alessandro Tacchini,[a] Vincenzo Testa,[b]
Massimo Trifoglio,[a] Antonio Tutone,[b] Stefano Vercellone,[e]
Roland Walter,[g] and for the ASTRI Project[‡]

[a]INAF/OAS Bologna, Bologna, Italy
[b]INAF/OAR Roma, Monte Porzio Catone, Roma, Italy
[c]ASI/SSDC Roma, Roma, Italy
[d]Università degli Studi di Perugia, Dipartimento to di Fisica e Geologia, Perugia, Italy
[e]INAF/OA Brera, Milano, Italy
[f]AC3E, Universidad Tecnica Federico Santa Maria, Valparaíso, Chile
[g]University of Geneva, Geneva, Switzerland
[h]INAF/OACT Catania, Catania, Italy
[i]INAF/IASF Palermo, Palermo, Italy
[j]INAF/IAPS Roma, Roma, Italy
[k]INAF/IASF Milano, Milano, Italy

ABSTRACT. The Astrophysics with Italian Replicating Technology Mirrors (ASTRI) Mini-Array is an international collaboration led by the Italian National Institute for Astrophysics (INAF) and devoted to imaging atmospheric Cherenkov light for very-high γ-ray astrophysics, detection of cosmic-rays, and stellar Hambury-Brown intensity interferometry. The project is deploying an array of nine dual-mirror aplanatic imaging atmospheric Cherenkov telescopes of 4-m class at the Teide Observatory on Tenerife in the Canary Islands. Based on SiPM sensors, the focal plane camera covers an unprecedented field of view of 10.5 deg in diameter. The array is most sensitive to γ-ray radiation above 1 up to 200 TeV, with an angular resolution of 3 arcmin, better than the current particle arrays, such as LHAASO and HAWC.

[*] Address all correspondence to Andrea Bulgarelli, andrea.bulgarelli@inaf.it

[†] Joseph (Joe) Hillary Schwarz passed at the end of August 2023. The present work is dedicated to him, gratefully.

[‡] http://www.astri.inaf.it/en/library/.

We describe the overall software architecture of the ASTRI Mini-Array and the software engineering approach for its development. The software covers the entire life cycle of the Mini-Array, from scheduling to remote operations, data acquisition, and processing until data dissemination. The on-site control software allows remote array operations from different locations, including automated reactions to critical conditions. All data are collected every night, and the array trigger is managed post facto. The high-speed networking connection between the observatory site and the Data Center in Rome allows for ready data availability for stereoscopic event reconstruction, data processing, and almost real-time science products generation.

Keywords: ASTRI Mini-Array; Cherenkov telescopes; software architecture; software engineering

Paper 23044G received Apr. 6, 2023; revised Nov. 28, 2023; accepted Dec. 22, 2023; published Jan. 23, 2024.

1 Introduction

Astrophysics with Italian Replicating Technology Mirrors (ASTRI)[1,2] is a project aimed at developing the next generation of imaging atmospheric Cherenkov technique (IACT) telescopes for ground-based γ-ray astronomy in the energy band between 1 and several hundred TeV. It was initially funded as a flagship project by the Italian Ministry of University and Research and is now one of the most significant ground-based astronomy projects led by INAF, focusing on both technological and scientific advancements.

The ASTRI-Horn prototype telescope has a diameter of 4 m and is located at the Serra la Nave site on the slopes of Mount Etna in Sicily. It is managed by INAF-OACt and was named after Guido Horn D'Arturo, an Italian astronomer who invented telescopes with tiled mirrors. This telescope is the first of its kind in Cherenkov astronomy, with a wide field of view of 10 deg and a compact, aplanatic, two-mirror optical configuration of the Schwarzschild-Couder type. In addition, it has a camera equipped with SiPM silicon sensors.

The ASTRI collaboration has gained valuable experience from the implementation of the first phase of the project and is currently working on the second phase which involves the installation of nine Cherenkov telescopes, known as ASTRI Mini-Array, at the Teide Observatory in Tenerife, Canary Islands, Spain. These telescopes are spaced ∼250 m apart and are like ASTRI-Horn but with some improvements. The project is being supported on-site by the Istituto Astrofisico de Canarias (IAC) and the Fundacion Galileo Galilei (FGG), which is governed by INAF. Other international institutions, such as Universidade de São Paulo (USP) in Brazil, North-West University South Africa, Université de Geneve in Switzerland, are also contributing at various levels. The ASTRI-Horn and ASTRI Mini-Array telescopes are prototypes of the small-size telescopes (SST) that will be installed at the southern site of the Cherenkov Telescope Array Observatory (CTAO) in Chile, at Paranal, which is expected to be operational by 2027.

The ASTRI Mini-Array aims to conduct stereoscopic observations in Cherenkov light from 2025 onwards.[3] Due to the IACT,[4] it is possible to infer the direction and spectrum of γ-ray photons with energies in the range from a few hundreds of GeV to 200 TeV and beyond arriving at the Earth from astrophysical sources. This will be the first time such observations are performed with wide-field telescopes. The focus will be on studying astronomical sources. Due to the precise angular and energy resolutions of the ASTRI Mini-Array, it will complement the high-altitude direct particle detectors at the northern site, such as LHAASO and HAWC. The latter are already monitoring the sky in the same band.

The ASTRI telescopes optics system is based on a primary mirror made of reflecting segments and a monolithic secondary mirror, arranged in a proper configuration. The Cherenkov UV-optical light produced by atmospheric particle cascades (air-showers), initiated by the primary γ-ray photons entering the atmosphere, is focused onto a compact camera (just 50 cm diameter) with a large (10.5 deg) field of view, due to a small plate scale. The camera is a fast (tens ns timescale) SiPM system developed by INAF adopting the CITIROC ASICS.[5]

The collected data are recorded and the array trigger is managed off-line, combining together the data taken by the different telescopes, after that a proper time stamp is granted by the White Rabbit common timing system.[6] Appropriate data analysis methods are employed to reduce the level of the background and allow efficient detection of γ-rays coming from astrophysical sources.

Besides the γ-ray scientific program, the ASTRI Mini-Array will also perform stellar Hambury-Brown intensity interferometry studies and cosmic rays detection in the PeV region due to the Cherenkov light analysis. To perform Stellar Hambury-Brown intensity interferometry observations[7] is possible because each telescope of the ASTRI Mini-Array will be equipped with an ad hoc, very fast camera for intensity interferometry. The ASTRI Mini-Array layout, with its very long baselines (hundreds of meters), will allow us to obtain angular resolutions down to 50 micro-arcsec, making it possible to reveal details on the surface of bright stars and of their surrounding environment and to open new frontiers in some of the major topics in stellar astrophysics.

The measurements of cosmic rays are also possible because 99% of the observable component of the Cherenkov light has a hadronic nature. Even if the main challenge in detecting γ-rays is to distinguish them from the much higher background of hadronic cosmic rays, this background, recorded during normal γ-ray observations, can be used to perform measurements and detailed studies of the cosmic rays themselves.[8]

The ASTRI Mini-Array telescopes, including the Cherenkov Camera,[5] are an updated version of the ASTRI-Horn Cherenkov Telescope[9] operating at Serra La Nave (Catania, Italy) on Mount Etna. The software developed by INAF for the ASTRI-Horn telescope, including development, testing, and production environments, is partially reused in the ASTRI Mini-Array context.

The ASTRI Mini-Array Software System presented in this paper manages observing projects, observation handling, remote array control and monitoring, data acquisition, archiving, processing and simulations of the Cherenkov and intensity interferometry observations, including science tools for the scientific exploitation of the ASTRI Mini-Array data. The ASTRI Mini-Array Software System[10] is under development by INAF teams and other Italian research institutions (including other public research institutions, such as the University of Perugia and INFN), foreign institutes (University of Geneva), and private partners (the Advanced Center for Electrical and Electronic Engineering—AC3E—at Santa Maria University, Valparaiso, Chile). INAF is in charge of software management and coordination, requirements specifications, top-level architecture definition, development, integration, verification, validation, and deployment of the overall software, with AC3E as an external contractor to manage some of these activities.

1.1 ASTRI Mini-Array System

The ASTRI Mini-Array system[1,2] is geographically distributed in three main sites. The Array Observing Site (AOS) at the Teide Observatory is operated by the Instituto de Astrofisica de Canarias (IAC), where the nine telescopes and the remaining observing site system are under installation; the AOS includes a data center for computing and networking resources. Some array operation centers (AOCs) are planned, each equipped with a control room. These AOCs are located remotely at various INAF institutes in Italy, at the IAC facilities in La Laguna (Tenerife), and at the Teide site for use during the installation and commissioning phases. A primary control room will allow the operator to supervise and carry out the scheduled observations and calibrations during the night, commanding the ASTRI Mini-Array, while an astronomer on-duty (AoD) supports and manages the observations; additional control rooms can allow monitoring the night operations. Finally, the ASTRI Data Center is in Rome for data archiving, processing and quick-look, simulation, and science user support.

ASTRI Mini-Array Software System runs at the AOS (the on-site software is called supervisory control and data acquisition, SCADA) and in the ASTRI Data Center in Rome (the off-site software).

The on-site software controls and monitors the observing site system, the site service system, and the safety and security system installed at the AOS. The array operator and AoD can remotely connect to the on-site software from AOCs through a web interface named operator human machine interface (operator HMI) that allows remote access, monitoring, and control of the on-site systems.

The observing site system is composed of all subsystems aimed at performing the observation:

1. The array system comprises nine telescopes with their assemblies, including the two main scientific instruments permanently mounted on each telescope: the Cherenkov Camera and the Stellar Intensity Interferometry Instrument.[11,12] In addition, an optical camera can be mounted on each telescope and is used for calibration and maintenance activities. Each telescope has a pointing monitoring camera installed on the rear of the secondary mirror support structure to obtain astrometric calibrated field-of-view of the region pointed by the telescope.[13]
2. The atmosphere characterization system, which includes three instruments: (i) the light detection and ranging (LIDAR) to study the atmospheric composition, structure, clouds, and aerosols through the measurement of the atmospheric extinction profile; (ii) three sky quality meters, to measure the brightness of the night sky in magnitudes per square arcsecond, two mounted on two telescopes, plus one with the all-sky camera, an instrument that monitors the cloud coverage; and (iii) the UVSiPM,[14] a light detector that measures the intensity of electromagnetic radiation in the 300 to 900 nm wavelength range; the analysis of the UVSiPM data is used mainly to evaluate the level of the diffuse night sky background (NSB).
3. The array calibration system, with one device, the illuminator, a portable ground-based device that allows determining the response efficiency of the telescopes. The illuminator is designed to uniformly illuminate the telescope's aperture, either with a pulsed or continuous reference photon flux; the National Institute of Standards and Technology calibrated photodiode monitors the absolute intensity.

The site service system is composed of all subsystems that provide services required to support the observing site system. The main subsystems are

1. the power management system, including centralized uninterruptible power supply system, to provide power to the entire ASTRI Mini-Array on-site system;
2. The telescope service cabinets serve as a point of connection of each of the nine ASTRI telescopes to the main ASTRI Mini-Array electrical, networking, safety, and time synchronization system;
3. The information and communication technology (ICT) system[15] includes the computing and networking infrastructure and all on-site and off-site system services to control and monitor the array, archive and analyze the scientific and engineering data. The ICT also includes the time synchronization system, which synchronizes with a sub-ns precision the Cherenkov Cameras to tag the Cherenkov events properly and is composed by a White Rabbit master switch to distribute the timing, a GPS antenna, and the master clock;
4. The environmental monitoring system for the evaluation of the environmental conditions: (i) two weather stations, (ii) humidity sensors, (iii) rain sensors, for prompt detection of rain, acquired at 2 Hz, and (iv) an all-sky camera for monitoring cloud coverage both during day and nighttime.

Finally, the safety and security system does not depend on any other site-installed system other than power. The functional safety actions are the detection of interlock requests and emergency stops. In case of hazardous faults, the system interlocks any other system that could be in a hazardous situation because of that fault. The safety and security system will be connected to the site emergency stop (E-stop) system that, if activated, shall trigger an emergency stop function. Emergency stop devices must be a backup to other safeguarding measures, not a substitute. E-stop devices shall be appropriately distributed throughout the site (e.g., local control room, service cabinet) to facilitate a quick activation from different locations in an emergency.

Each hardware assembly has a local control system, i.e., a hardware/software system used to switch-on/switch-off, control, configure and get the status, monitoring points, and alarms of all parts of the assembly: the related software is called local control software (LCS). Each LCS could have a local engineering human machine interface (engineering HMI). LCSs can be delivered as part of an externally contracted subsystem or developed by the INAF team (e.g., for subsystems developed by INAF such as the optical camera and the UVSiPM).

Each LCS implements an interface to the ASTRI Mini-Array Software System based on the IEC 62541 standard for the OPC Unified Architecture protocol (OPC-UA).[16] It is one of the most important communication protocols for industry 4.0 and the internet of things. OPC-UA allows access to machines, devices, and other systems in a standardized way and enables similar and manufacturer-independent data exchange. An interface control document describes the interface between an LCS and the on-site software. Two subsystems use a different protocol: the on-site ICT system uses a simple network management protocol (SNMP) and the power management system uses a MODBUS protocol.[17]

The INAF team has in charge also the development of the assembly, integration, and verification (AIV) software used during the AIV activities, which could be connected with an LCS via the OPC-UA interface (see Sec. 4.4.7).

2 Observing Cycle

The ASTRI Mini-Array observing cycle is the main driver for developing the ASTRI Mini-Array software architecture. The ASTRI Mini-Array Software System is envisioned to handle the observing cycle, i.e., the end-to-end control and data flow, and the information and operations required to conduct all tasks from the time an observing project (a description of a scientific project to observe a target) is created until the resulting data are acquired and analyzed.

The main actors that interact with the software system are the following:

1. The science user performs observations related to the observing projects and analyzes science data after the completion of the observations;
2. The support astronomer prepares the long-term and short-term observation plans;
3. The operator is responsible for supervising and carrying out scheduled observations and calibrations during the night;
4. The AoD supports and supervises the observations during the night from a scientific perspective;
5. The archive manager is responsible for the quality and integrity of the data;
6. The configuration manager keeps track of the configuration of all instruments, part replacements, and assemblies configurations;
7. The maintenance engineer manages and executes maintenance activities and conducts on-site preventive and corrective maintenance tasks;
8. The expert operator is responsible for technical operations and AIV activities and is an expert in one or more assemblies or subsystems of the ASTRI Mini-Array system.

A schematic representation of the global information flow is given in Fig. 1, where the observing cycle's main phases and related functions are shown. The observing cycle is divided into four main phases: (i) observation preparation, (ii) observation execution, (iii) data processing, and (iv) dissemination.

The observation preparation is the first phase of the observing cycle. The observing cycle initiates by a science user submitting an observing project. Once the ASTRI Mini-Array Science Team has selected and approved a list of observing projects, the support astronomer, with the help of an observation scheduler (see Sec. 4.4.2) tool, turns them into a list of scheduling blocks (SBs) containing all the information required to perform the corresponding observations, including time constraints and telescope constraints and configuration. SBs are divided into observing blocks (OBs), i.e., the smallest sequence of observing instructions that can be scheduled, and that depends on the observation mode chosen by the submitter of the proposal. For example, suppose the Wobble observation mode is chosen as the observing mode. In that case, a single SB will be divided into a sequence of OBs, which foresee a calibration run as the first OB and then 2 or 4 OBs with alternate wobble target positions. SBs are scheduled in long- and short-term observation plans and stored in the archive system. The short-term observation plan (the list of SBs that must be observed during the night) is transferred on-site.

The next step of the observing cycle is the observation execution. The central control executes the short-term observation plan of the observing night, carrying out setups (with an appropriate set of configuration parameters), calibrations, and target observations necessary to ensure that the

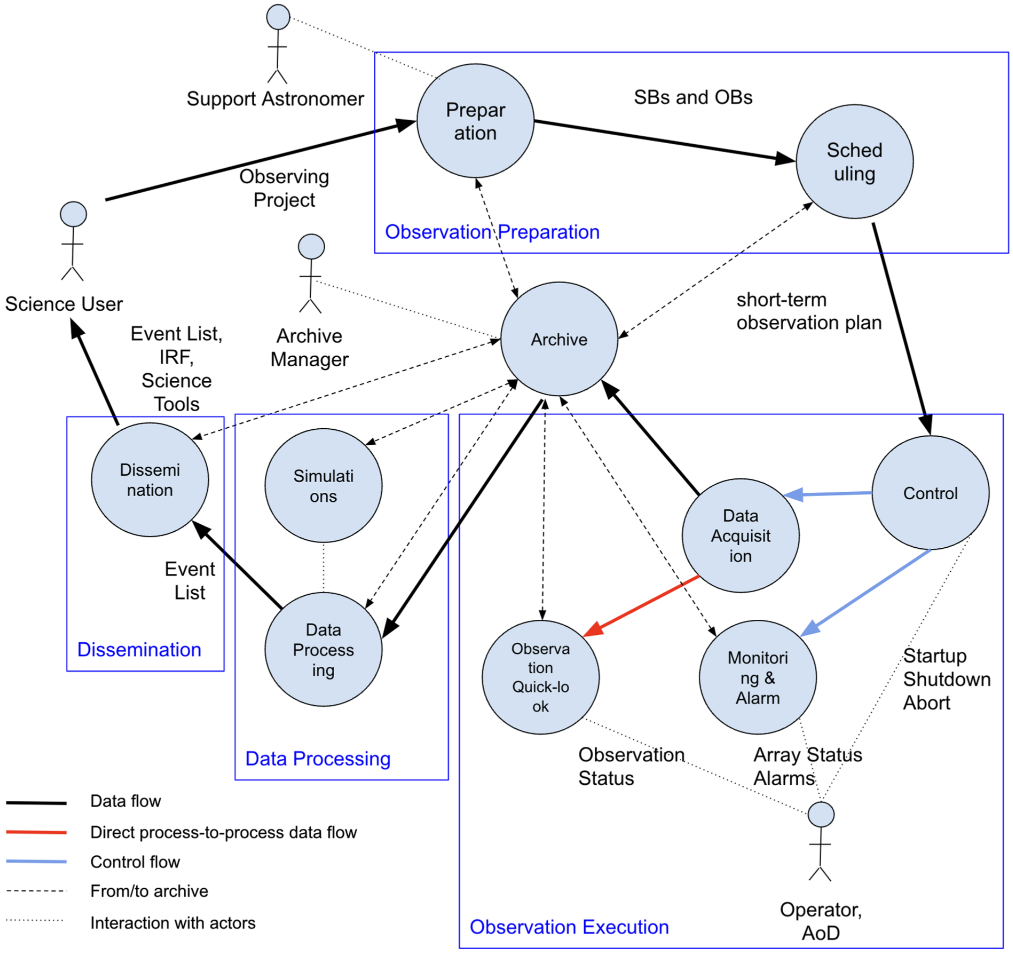

Fig. 1 ASTRI Mini-Array data and information flow (schematic) with the four main phases. The outer solid black and red lines show the logical data flow, whereas the solid blue lines are the control flow. Direct process-to-process communication is indicated with a red line. The science user initiates the observing cycle with an observing project and gets the final results for the scientific exploitation of the observations. The archive manager is responsible for the quality and integrity of the data. The dashed lines directed to/from the archive indicate that (a) all data are saved and can be retrieved from the archive and (b) the archive may handle the physical data flow. The operator and the AoD are responsible for the nightly operations. The main phases are described in the text.

acquired data are correctly calibrated and used in the construction of the final data product. The array operator remotely supervises operations at the ASTRI Mini-Array AOS via a remote operator HMI. The on-site software starts the array elements, checks the array's status, assesses the environmental conditions and atmosphere characterization (e.g., NSB level), performs the array calibration, and checks the observation data quality. The array operator can also manually change the schedule, check the status of assemblies, and administer other resources. Changes in environmental conditions, atmosphere characterization, or array status can change the kinds of observations that can be carried out; SBs are scheduled or stopped considering current conditions.

At the end of an OB, the data are transferred off-site; this starts the data processing phase. The data processing produces calibrated and reconstructed data (the final event list), applying necessary corrections. Monte Carlo simulations are performed to optimize the reconstruction of the Cherenkov events. Automated scientific analysis is performed on reconstructed data. If an external science alert is detected, the short-term observation plan is modified to follow up on the interesting astrophysical multi-messenger (GW or neutrino) and multi-wavelength events.

Data and science tools are distributed for a scientific analysis of the observing projects to the science users: this is the data dissemination phase. Science tools can be used to produce images and spectra and detection of γ-ray sources. High-level data and data products [event lists and instrument response functions (IRFs)] are released to the ASTRI Mini-Array Science Team.

Storing all persistent information in the archive system makes the system less coupled so that these phases can work independently as long as they maintain the information flow to and from the archive system.

3 Main Requirements

To reduce overall operation costs and workforce, the following top-level requirements are considered for the definition of the software architecture:

1. The ASTRI Mini-Array System shall be controlled and monitored by software running on-site with the telescopes.
2. The ASTRI Mini-Array will become an open observatory after the first 4 years of operations.
3. No human presence is foreseen at the site during the nights. The ASTRI Mini-Array System shall be operated from AOCs available from different locations, including one at the AOS. Only one AOC shall control the array, while others shall be restricted to a read-only mode suitable for monitoring.
4. The ASTRI software shall allow the science team to define the scientific targets based on their visibility and the priority assigned to each science program. The long- and short-term observation plans shall be prepared and validated in advance with the help of suitable tools.
5. AoD of the ASTRI Mini-Array should have the capability, either manually or through an automated software system, to select, prepare, and execute target of opportunity (ToO) observations during the night.
6. The on-site software shall be able to automatically execute the whole sequence of operations to perform an observation.
7. A quick look of data at a single Cherenkov camera and intensity interferometry detector level shall be possible on-site by an online observation quality system.[18]
8. The on-site software shall be able to react to environmental critical and survival conditions automatically to put the array system in a safe state.
9. The amount of data storage installed at the observing site shall be adequate to guarantee no loss of technical and scientific data in case of a lack of connection to the wide-area network. In particular, the on-site storage shall be able to maintain data for at least 7 days, including raw scientific data, monitoring, logging and alarm data, and online observation quality system data products.
10. All data shall be transferred to the remote data center in Rome, Italy, at the end of each run, where they will be permanently archived.
11. Any search for Cherenkov events detected in coincidence by multiple telescopes (stereoscopic event reconstruction) shall be performed off-line at the Rome Data Center.
12. All data processing shall be done off-line at the Rome Data Center, including the historical analysis of monitoring and logging data.
13. All highest-level data products associated with observing projects produced by the off-line data processing shall be validated, archived, and made accessible to the ASTRI Science Team.
14. The ASTRI Science Team shall provide dedicated science tools for scientifically exploiting the ASTRI Mini-Array data.

The ASTRI Mini-Array software developed by the ASTRI collaboration and used during operations, with only a few exceptions regulated by industrial contracts, is primarily governed by the Lesser General Public License (LGPL) from the Free Software Foundation. The software will become open-source as soon as a future fully operational version is released.

4 General Software Architecture

4.1 4+1 Architectural View Model

The primary goal of software architecture is to illustrate the organization of the software system, delineate its structural components and their functionalities, and integrate these components into broader subsystems. The architectural approach used by ASTRI Team is the 4+1 view architectural model[19] illustrated in Fig. 2 and consists of looking at the system through different views, represented with unified modeling language (UML)[20] diagrams: (i) the use-case view describes the system's interaction with actors developing use cases. A use case is a list of actions or event steps typically defining the interactions between an actor and a system to achieve a goal. The actor can be a human or other hardware or software systems; (ii) the logical view is a functional decomposition of the system with the description of the global information flow based on the analysis of use cases and data models; (iii) the process view deals with the dynamic aspect of the system; (iv) the implementation/development view represents the detailed design of the implemented system; (v) the physical/deployment view depicts the system from a system engineer's point of view: the physical view is more concerned with the system's physical layer, and the deployment view deals with allocating computing resources on physical nodes. It concerns the topology of software components on the physical layer and their physical connections.

The ASTRI Team adopted the 4+1 view model because this allows a deep integration of the domain experts (e.g., scientists, instrument developers) with the software developer team. For example, this allowed them to actively participate in the requirement definition, developing use cases directly to integrate their knowledge of γ-ray astrophysics, astronomical observatories, instrument development and operations in the overall definition of the software architecture. Experts and scientists have also actively participated in the definition of logical and process views.

4.2 Requirement Engineering

The main purpose of the requirement engineering process is to produce functional and quality (a.k.a. non-functional) requirements. The requirement inception is the first step to collect the requirements from users and other stakeholders to

1. understand the workflow, starting with user expectations;
2. maintain costs within a chosen envelope, deciding what to build, what the system must do, how it must behave, the properties it must exhibit, the qualities it must possess, and the constraints that the system and its development must satisfy;
3. maps the functionalities to the science requirements.

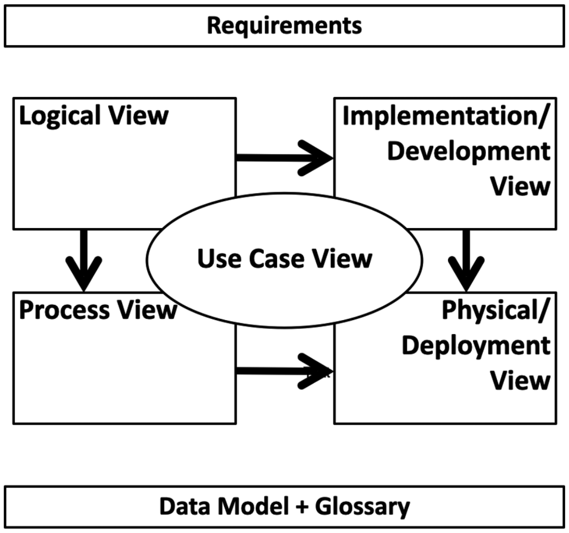

Fig. 2 Illustration of the 4+1 architectural view model with requirements, data model, and glossary to complement the information.

A requirement inception process is a challenge because there are many different problems during this phase:[21] (i) problem of scope: the user specifies technical details, and the boundary of the system is not well defined; (ii) problem of understanding: the users do not have a complete understanding of the problem domain, have trouble communicating needs to the system/software engineers, omit information that is believed to be "obvious," specify requirements that conflict with the needs of other customers/users, and have a glossary with terms with different meanings; (iii) requirements volatility: the requirements change over time.

The development of some views is part of this process; in particular, the use case, logical, and process views are used to define the scope and the main functions of the software system. An initial definition of some top-level requirements (listed in Sec. 3) has been provided to address the problem of understanding, coupled with a glossary and a high-level definition of the data model of the ASTRI project. In this way, many ambiguities have been removed from the beginning of the project facilitating the requirement inception phase.

To keep the problems depicted above (scope, understanding, and volatility) under control, we have adopted an iterative process for the definition of a set of top-level software documents to develop the views "so that solutions can be reworked in the light of increased knowledge."[22]

4.3 Top Level Software Documents

The content of the top-level software documents (also called software system engineering documents) is summarized in this contribution. These documents passed a Concept Design Review (CoDR, see Sec. 5.3) with a panel of external reviewers in June 2020. These documents include (i) the top level use-case document, (ii) the top level software architecture document, (iii) the top level data model document, (iv) the product breakdown structure (PBS), and (v) a global glossary at project level.

The main inputs for defining these documents were the ASTRI science and system requirements, the ASTRI operation concept,[2] and the ASTRI science use cases.

The top level data model document provides a conceptual view of the ASTRI Mini-Array data model, describing data products, data models and their relationship, referring to data streams in architectural diagrams without ambiguity, and defining a short identifier for the data product. The concepts and definitions described in this document and the glossary are references for all software documents developed by the ASTRI team.

The top level use cases document captures the greatest possible number of stakeholder's points of view analyzed during the requirements inception phase. This document contains observation-related use cases that describe how to perform observations from the proposal to the scientific exploitation of the acquired data from a user's point of view and the commonalities of all the science-related use cases, according to the observing cycle described in Sec. 2. This category includes calibration and other technical-related use cases. This document covers the use-case view of the system and is the starting point for the development of detailed use-case documents at the subsystem level. The iterative process adopted by the ASTRI team allowed using these use cases as a high-level process view, including human actors and some top-level system actors defined in the top level architecture document.

The top level architecture document provides a comprehensive architectural overview of the ASTRI Mini-Array Software System and the hardware installed at Teide from a logical perspective, providing a complete functional decomposition and the main requirements of the software. This document covers the logical view, partially the process view, and the deployment view. This document depicts various aspects of the software using different views and describes the most significant architectural decisions.

Use cases coupled with the functional view provide a complete description of the functional requirements of the software.

The functional decomposition described in the top level architecture has been used to develop the whole PBS of the software system, used to manage interfaces and to define the specification tree, i.e., the definition of the hierarchical relationship of all technical aspects of the software system and is the basic structure to perform requirements traceability.

The PBS has also been used to define the project's work breakdown structure (WBS), allowing an organization of the work based on the customer-supplier relationship described in Sec. 5.1.

The top-level documents serve as the foundation for a more comprehensive requirement elicitation phase. The requirement elicitation phase comes after inception and involves gathering detailed requirements. It aims to uncover specific needs, features, and constraints by interacting with stakeholders and users. This phase involves the development of detailed use-case documents and software requirements for each software subsystem within the ASTRI Mini-Array software architecture (refer to Sec. 4.4). In addition, these requirements, along with the top-level documents, are used to create detailed design documents for each software subsystem. This process ensures traceability between the subsystems and the top-level use cases and architectural elements, effectively constructing the complete specification tree of the ASTRI Mini-Array Software System.

4.4 ASTRI Mini-Array Software Main Systems

The general architecture of the ASTRI Mini-Array Software System is derived from the use cases, data models, and data flow definitions and consists of the top-level systems described in this section. Figure 3 shows the context view with the main software systems, which are the archive system, the science support system, the SCADA system, the data processing system (DPS), the simulation system, the on-site startup system, and the AIV and engineering software. The following sections provide an overview of these systems, with a short description of the main functionalities and a link with the observing cycle phases.

4.4.1 *Archive system*

The archive system (see Fig. 4 with the connected data models) provides a central repository for all persistent information of the ASTRI Mini-Array, such as observing projects, observation plans, raw and reduced scientific data, monitoring data, system configuration data, logs of all operations and schedules. The main archives are

1. The bulk archive stores data and calibration from scientific instruments;
2. The science archive manages observing project, observation plans, the science data model (SDM) (see Sec. 4.5), and the scientific results;

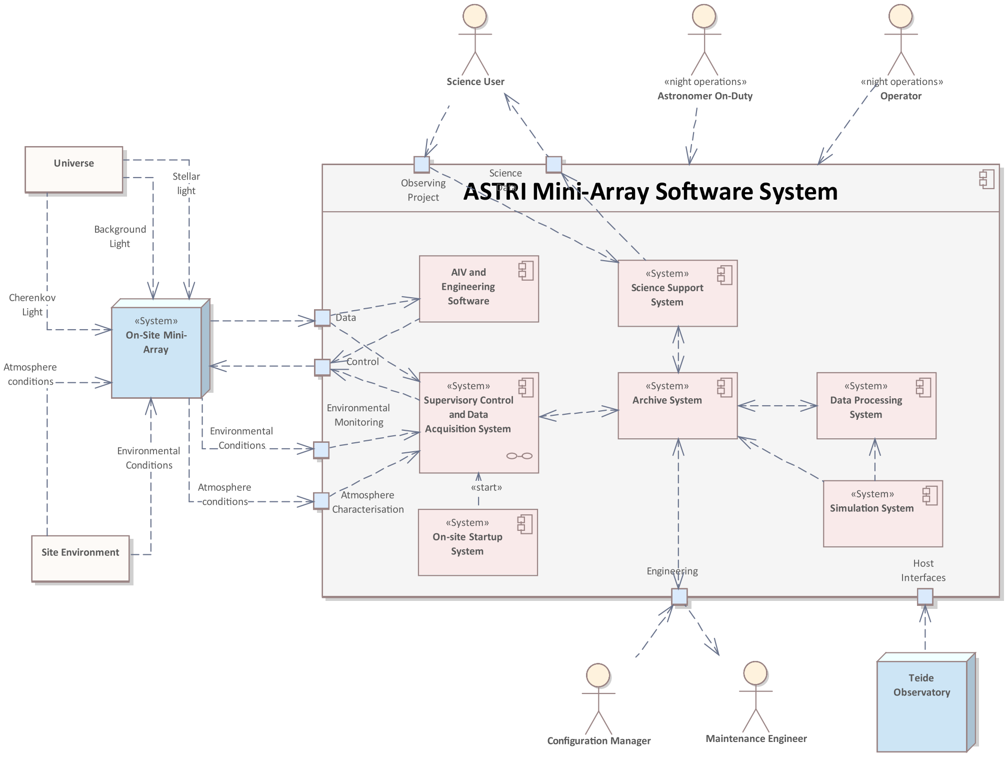

Fig. 3 Context view of the ASTRI Mini-Array Software System and all software systems.

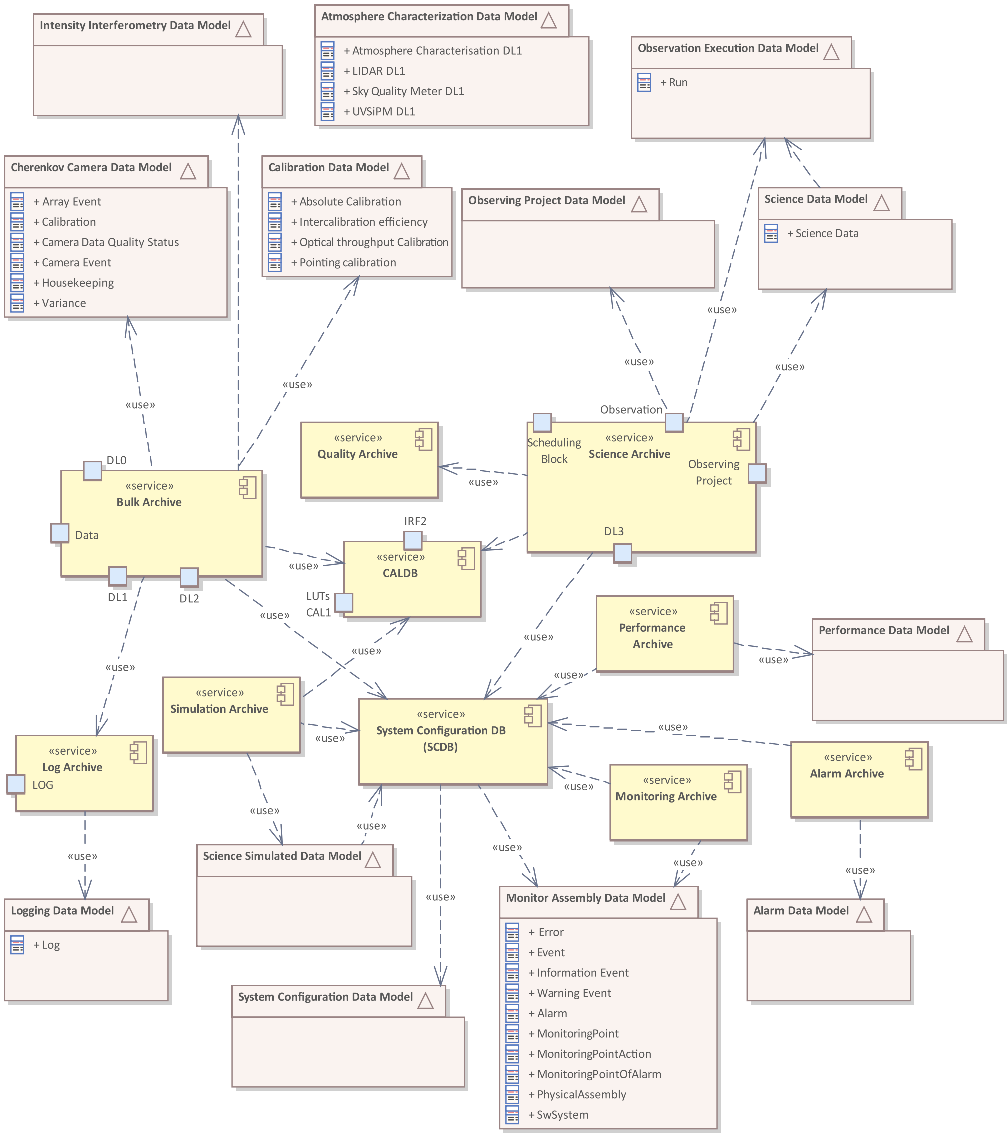

Fig. 4 Archive system and the relationship of each archive with the data models and between archives.

3. The system configuration database stores the configuration of the ASTRI Mini-Array System;
4. Monitoring archive, log archive, alarm archive store logs, monitoring points, and alarms produced by hardware and software on-site subsystems. The monitoring archive also stores the products of the environmental monitoring system and atmosphere characterization system;
5. The quality archive stores the Cherenkov and intensity interferometry observation quality checks during the observation;
6. The CALDB is the calibration database and stores IRFs and other instrumental and precomputed quantities;
7. The simulation archive contains all the Monte Carlo simulated events;
8. The performance archive contains reduced engineering data used to perform mid- and long-term performance and predictive studies.

4.4.2 *Science support system*

The science support system manages the observing projects, and the observation plans preparation, the management of science alert events, the dissemination of scientific data, and the science tools for their analysis. It is the main interface for science users to the ASTRI Mini-Array

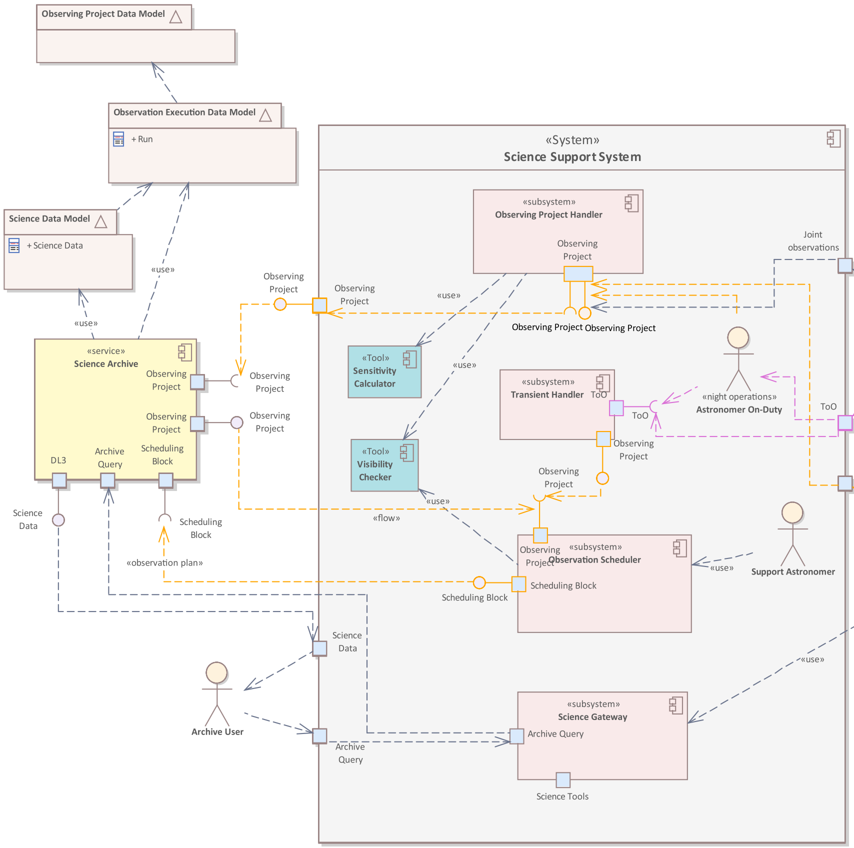

Fig. 5 Science support system component diagram with the science archive and related data models and external interfaces.

system. It provides them with an easy-to-use science support system HMI for the detailed specification of observations. The main products generated by this system are the observation plans. The science support system also contains the science gateway. The science user uses the web interface to access high-level science-ready data and data products delivered by the DPS. This system supports the observation preparation and the dissemination phase of the observing cycle. The main functions are (see Fig. 5):

1. The observing project handler is used to submit observing projects, to store the long-term observation plans and to select the short-term observation plans for the next night;
2. The transient handler handles external science alerts and follow-up observations; this is the interface between the ASTRI Mini-Array system and the external facilities/brokers that will provide real-time science alerts on astrophysical transients;
3. The observation scheduler supports the preparation of long-term observation plans, short-term observation plans, and observing projects;
4. The science gateway to retrieve science-ready data, science tools, and tools to support the observing project preparation.

The transient handler is responsible for submitting a new observing project to the observation scheduler whenever an interesting external alert is received and flagged as observable. Upon receiving this trigger from the transient handler, the observation scheduler will generate a new short-term observation plan for the ToO observation and provide it to the SCADA/central control that will be triggered to execute the new short-term observation plan.

4.4.3 *Supervisory control and data acquisition system*

The SCADA system controls all operations at the AOS. SCADA's central control system interfaces and communicates with all assemblies and dedicated software installed at the site.

It is responsible for the execution of the short-term observation plan to perform observations. SCADA shall be supervised by the operator but performs the operations in an automated way. It shall provide scientific data, logging, monitoring, alarm, and online observation quality information to help assess data quality during the acquisition. This system supports the day and night observation execution and maintenance phases. The main functions (see Fig. 6) are:

1. Central control system, developed by AC3E,[23] coordinates the sequence of operations, startups, shutdowns, configures, and checks the status of the on-site ASTRI Mini-Array systems. It gets and validates SBs of a short-term observation plan and executes the OBs interpreting the observing mode to command the telescopes and other subsystems. The data capture stores the information associated with the execution of an OB (see Sec. 4.5). The central control system is also composed of controls and collectors subsystems:

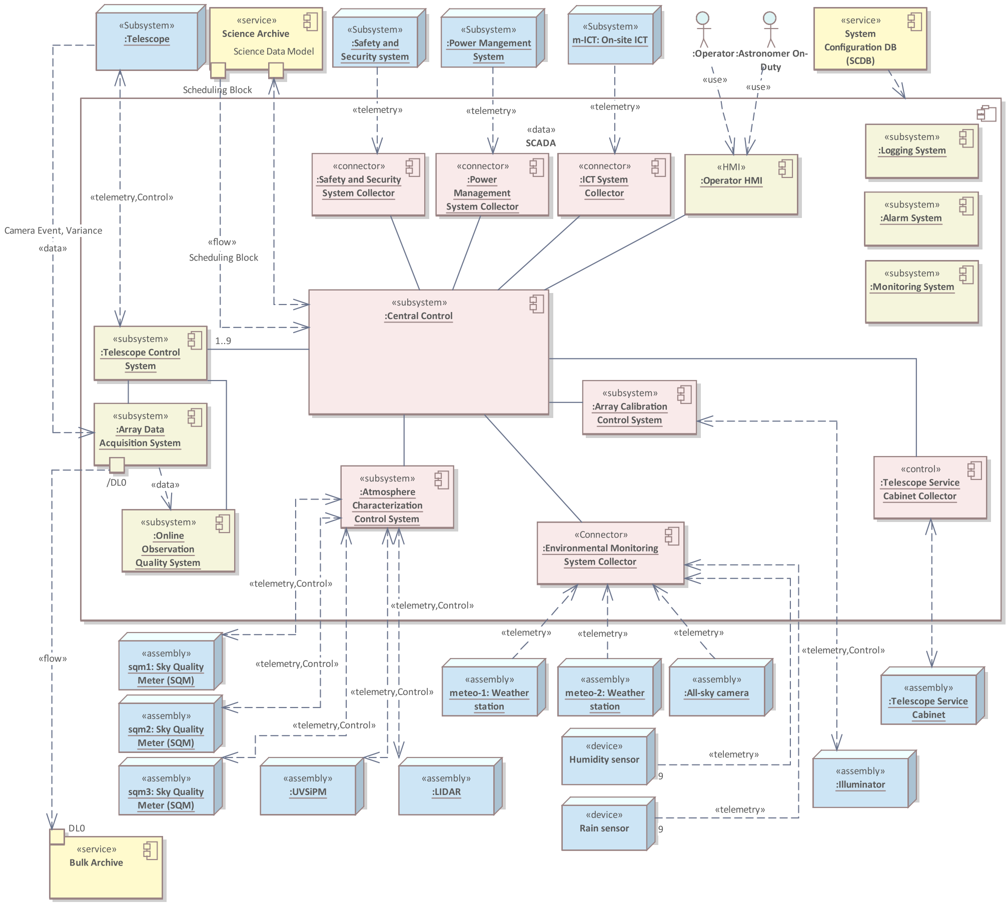

Fig. 6 SCADA component diagram. The logging system acquires logs from all systems. The monitoring system acquires monitoring points from all assemblies and software systems. The alarm system receives alarms from assemblies or software systems to display them to the operator. Only one telescope (and related subsystems) is shown, but there are nine independent chains of control, data acquisition, and quality checks. Not all connections are shown; in particular, the interconnections between control software/collectors and central control are not shown, and all connections between alarm system, monitoring system, and logging system are not shown. Light red and green components are the SCADA system, where the light red components are part of the central control system, the blue nodes are the ASTRI Mini-Array hardware assemblies, and the yellow components are part of the archive system. The ≪telemetry≫ stereotype represents monitoring points, alarms, errors, logs, and status information, ≪data≫ stereotype represents the data flow. The ≪control≫ stereotype represents the control flow. DL0 is the raw data generated by scientific instruments.

(a) Control systems are used to control, monitor, and manage alarms and the status of the telescopes (telescope control system,[24,25] developed by INAF based on the ASTRI-Horn experience), of the assemblies used to characterize the atmosphere (atmosphere characterization control system), and of the calibration system (array calibration control system);

(b) Collectors, to monitor and determine alarms and the status of environmental devices (environmental monitoring system collector), of the ICT system[15] (on-site ICT system collector), of the power system (power management system collector), of the safe and security system (safety and security system collector), and the telescope service cabinets (telescope service cabinet collector, one for each telescope);

2. Array data acquisition system,[26,27] developed by INAF, acquires Cherenkov cameras and stellar intensity interferometry instruments data that are saved in the bulk archive;
3. Online observation quality system,[18] developed by INAF, evaluates during the observations the data acquired by the instruments to obtain the status of the observations at a single telescope level. The results are saved in the quality archive;
4. Logging system, monitoring system, and alarm system,[28] developed by INAF, monitor the overall assemblies of the systems through the acquisition of environmental, monitoring and logging points and alarms from instruments and generates status reports or notifications to the operator. Data are saved in the logging archive, monitoring archive, and alarm archive, respectively;
5. Operator HMI, developed by the University of Geneva, is the user interface for the operator, including an operator logbook to save logs of the observations during the night.

Each SCADA subsystem could provide an engineering HMI, i.e., a dedicated graphical user interface for development, troubleshooting, and test purposes.

SCADA is developed using the ALMA Common Software (ACS).[29] ACS[30] is a container component framework designed for distributed systems, with standardized paradigms for logging, alarms, location transparency, and support for multiple programming languages: Java, C++, and Python. ACS has been used successfully for the Atacama Large Millimeter Array (ALMA) Observatory, which manages an array of 66 antennas on the Chajnantor plateau in Chile. ACS has also been used for ASTRI-Horn and the Sardinia Radio Telescope[31] and is also used for CTA.[32] Most of Mini-Array's software developers in INAF are, therefore, familiar with the use of ACS.

4.4.4 *Data processing system*

The DPS[33] (see Fig. 7) performs the calibration of scientific data, data reduction, and analyses. It also checks the quality of the final data products. Its primary role is to process data retrieved from the archive system as soon as enough data have been acquired to make such reduction meaningful. Typically, processing will be performed on data sets arising from an SB. This system supports the observing cycle data processing phase.

The main functions are: (i) the stereo event builder,[34] to perform the off-line software stereoscopic event reconstruction of Cherenkov data; (ii) the Cherenkov data pipeline including the calibration software pipeline,[33] for data calibration, reconstruction, selection, and automated scientific analysis of Cherenkov data; (iii) the intensity interferometry data reconstruction, and (iv) scientific analysis pipeline,[11] for reconstruction and analysis of the Stellar Intensity Interferometry data.

4.4.5 *Simulation system*

The simulation system provides Monte Carlo simulated scientific data for developing reconstruction algorithms and characterizing real observations.

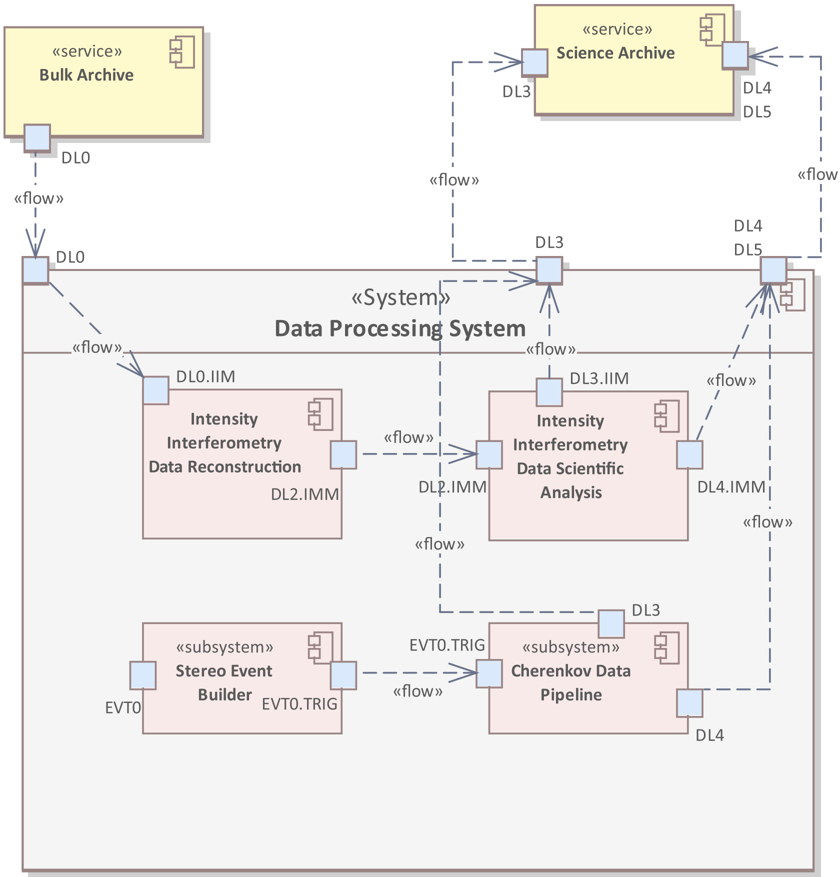

Fig. 7 DPS component diagram. DL0 is the raw data from scientific instruments, IMM is the intensity interferometry data, and EVT is the Cherenkov data. EVT0.TRIG is the Cherenkov data after the stereo array trigger. DL3 and DL4 are scientific products. Components are described in the text.

4.4.6 *On-site startup system*

The on-site startup system shall manage the sequence of the startup and shutdown of the on-site hardware systems that are mandatory for the startup of the telescopes and connect assemblies of the observing site system and of the site service system and the SCADA system.

4.4.7 *AIV and engineering software*

Each hardware assembly or subsystem could have an AIV and test software called AIV and engineering software. This software is connected with the LCS of a hardware subsystem via the OPC-UA interface. A local engineering HMI could be part of the AIV software.

4.5 Data Capture

The ASTRI Mini-Array System's software can be divided into telescope domain and the science domain. The telescope domain is instrument-centric, and the science domain is scientific observation-centric. The science support and DPSs are part of the science domain. SCADA is the bridge between the two domains.

The data capture, part of the central control system, takes the instrument-centric, time-ordered data stream, collects, and extracts those items needed in the science domain, and reorganizes them; it is responsible for collecting the metadata associated with the OB execution (the run), the data capture report. The data capture report is necessary to reduce and analyze the scientific data. The SDM describes the content of this metadata and provides links to the two domains. Figure 8 provides more details and links data capture and the data models in the telescope and science domains.

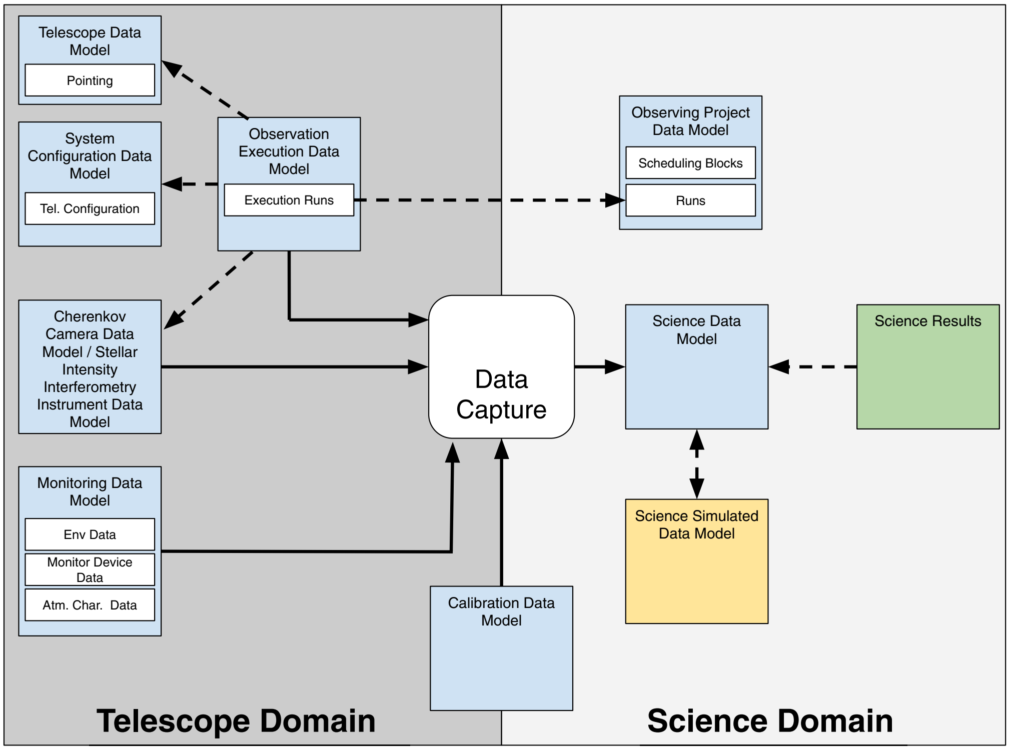

Fig. 8 The data capture and the data models in the telescope and science domains. A solid line indicates data flow, and dashed lines indicate referencing. Data flow streams from the left (upstream) to the right (downstream), although there could be some flow upstream to the data capture. The double referencing between the SDM and the science simulated data model means that simulations are linked with the corresponding SDM and vice versa. The information collected by the data capture is (i) observing data (Cherenkov camera data model, stellar intensity interferometry instrument data model), (ii) observing process description (observing project data model, observation execution data model, telescope data model, system configuration), (iii) monitoring data (environmental data model, atmosphere characterization data model, some data products of the monitoring data model), and (iv) some logging data not shown in the figure.

4.6 Operation of the MA Software System

This section provides a sketch of the architectural process view of the ASTRI Mini-Array Software System. Figure 9 summarizes the workflow and the main operation, and the numbering sequence is reported in the following paragraphs, where the workflow of the main software systems is described.

The science support system manages observing projects submitted by science user (1) and provides support to prepare the observation plan and associated SBs stored in the archive system (1.1).

At the beginning of the night, the validated short-term observation plan with all the relevant information (e.g., target and pointing coordinates, observing mode, OB duration) is uploaded from the science archive. The observation selection is performed automatically by the central control system or manually by the operator (2), which quickly cross-checks the array's status and environmental conditions through the operator HMI. The validated short-term observation plan for the night is retrieved to be executed manually or by setting the central control system in an automated way (2.1). The central control system manages the observation, fetching the current OB from the archive (2.2). The central control system configures the array assemblies and starts the array data acquisition system (2.3) and the online observation quality system (2.4). The alarm and monitoring systems are always running to have a full-time monitoring of the site.

When the hardware systems are ready, the operator starts the observation (3), and the central control system manages the list of OBs in an automated way. A run is the execution of an OB with an associated identifier. During the observation, the array data acquisition system acquires and saves raw data in the local bulk repository (3.1), while the online observation quality system

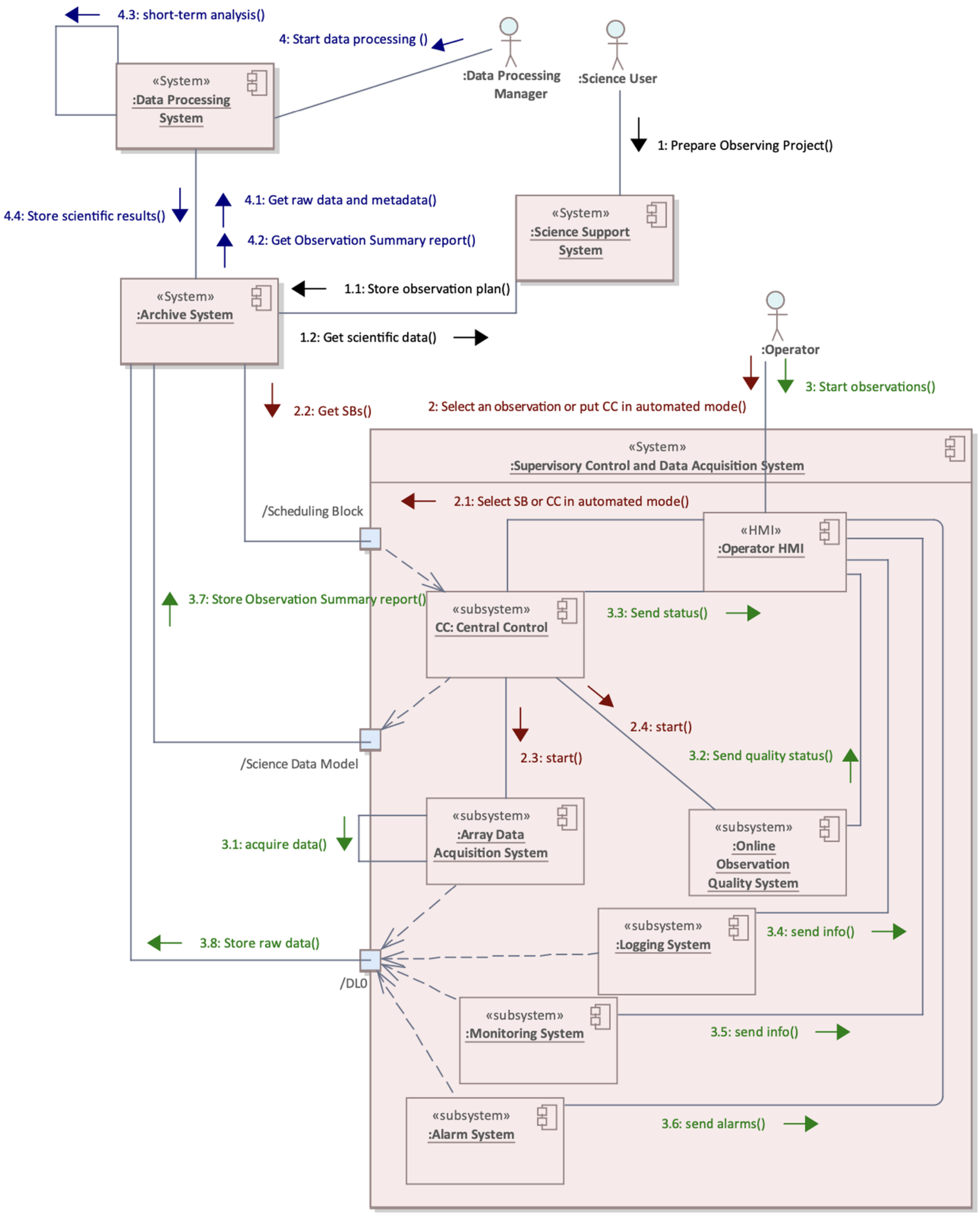
4.3: short-term analysis()
4: Start data processing ()
:Data Processing Manager
:Science User
«System» :Data Processing System
1: Prepare Observing Project()
4.4: Store scientific results()
4.1: Get raw data and metadata()
4.2: Get Observation Summary report()
«System» :Science Support System
«System» :Archive System
1.1: Store observation plan()
1.2: Get scientific data()
:Operator
3: Start observations()
2.2: Get SBs()
2: Select an observation or put CC in automated mode()
«System» :Supervisory Control and Data Acquisition System
2.1: Select SB or CC in automated mode()
/Scheduling Block
«HMI» :Operator HMI
3.7: Store Observation Summary report()
«subsystem» CC: Central Control
3.3: Send status()
2.4: start()
/Science Data Model
2.3: start()
3.2: Send quality status()
3.1: acquire data()
«subsystem» :Array Data Acquisition System
«subsystem» :Online Observation Quality System
«subsystem» :Logging System
3.4: send info()
3.8: Store raw data()
/DL0
«subsystem» :Monitoring System
3.5: send info()
«subsystem» :Alarm System
3.6: send alarms()

Fig. 9 Operations of the ASTRI Mini-Array Software System are described with a UML collaboration diagram. The numbered arrows indicate steps from the creation of an observing project through the control flow of the array until the acquisition, short-term data reduction, analysis and storage in the archive system. See text for more details.

focuses on ongoing problems in data quality (3.2) and sends a report to the operator HMI. During the observation, the data capture of the central control system prepares the observation summary report (see Sec. 4.5), i.e., collects all the engineering and auxiliary information needed by the DPS to reduce and analyze the raw scientific data.

During the observation, the operator checks the observation status through the operator HMI. The central control system sends information about the observation status (3.3), providing feedback to the operator. The logging system (3.4) and the monitoring system (3.5) send information to the operator HMI. The alarm system sends alarms to the operator HMI (3.6). The observation summary report is stored in the science archive (3.7), and the raw data are stored in the bulk archive (3.8).

At the beginning of the night also the DPS (4) is started. When a run is finished, the raw data (4.1) and the observation summary report (4.2) are transferred off-site in automated way.

A short-term analysis is performed at the end of the data transfer of a run (4.3) to produce preliminary science products, which are stored in the archive system (4.4). The operator checks some results of the DPS through the operator HMI.

The long-term data analysis is started when data are ready in the off-site archive. The DPS pipeline retrieves raw data and metadata (the observation summary report), as well as calibration coefficients (CAL1), look-up-tables, and IRFs, needed for Cherenkov data characterization and scientific analysis from the archive system and performs the complete data reduction. The DPS pipeline generates the final science-ready data and automatic science products and stores them in the archive system. Before the Cherenkov data analysis, a stereo event-building procedure employs an offline stereoscopic event reconstruction. This step is essential to exploit the stereoscopic capability of the array.

When science-ready data and science products are computed, they are made available from the archive system by the science support system to the science user (1.2).

5 Software Engineering Approach

The ASTRI software engineering office is part of the ASTRI system engineering activities of the ASTRI project office. It interacts with all ASTRI work packages by delivering coordination and integration services for developing ASTRI software. The ASTRI software engineer team, coordinated by a software system engineer, defines guidelines and planning for the ASTRI software development and deployment. These activities are coordinated with the ASTRI project office, which is responsible for all aspects of the project. The software engineering team coordinates activities with the ASTRI quality assurance team, the safety team, and the science team.

In the following sections, we describe the software life cycle and the organization of the developer teams, which is based on the tailoring of the European Cooperation for Space Standardization (ECSS)[35] integrated with Agile software development practices.

5.1 Customer-Supplier Relationship

The production of the ASTRI Mini-Array Software System requires the cooperation of several INAF work groups and external organizations that share the common objective of providing a software system that satisfies the overall scientific and technical requirements of the ASTRI Mini-Array. To organize the overall team, a customer-supplier relationship model has been adopted, where the customer accepts the software, having one or more software suppliers that must develop and deliver software according to the customer's requirements. This relationship is recursive, i.e., the customer could also be a supplier to a higher-software-level customer.

The ASTRI Mini-Array Software System's suppliers are INAF teams of different institutes and other research institutions, such as the University of Perugia, INFN, and the University of Geneva (Switzerland), and AC3E, which supplies part of the software of the SCADA system. INAF oversees software management and coordination, requirements specifications, and top-level architecture definition. Each supplier is responsible for developing, integrating, and verifying all sub-work-packages (sub-WPs) products. AC3E is also responsible for SCADA integration, verification of the integrated system, delivery and deployment, and supports the validation of the SCADA system.

This organization defines a complex customer-supplier chain. This requires overall project management following a structured approach throughout all stages of the software life cycle and at all levels of the customer-supplier chain. Management, engineering, and product assurance activities are integrated for the execution of the project.

The software system engineer is the top-level customer of the customer-supplier chain for the software. The software coordinator and the deputy software coordinator are the suppliers of the software system engineer that must provide the software systems identified in Sec. 4. Each software subsystem coordinator (SCADA, archive, simulation, data processing, science user support, and on-site startup) is a supplier for the software coordinator. Each software subsystem coordinator manages the effort provided by ASTRI developers, external contractors, and research institutes as suppliers.

5.2 Tools and Standards

The software is designed with the UML, requirements and design are managed and documented using the Enterprise Architect tool.[36] Released documents are managed using the DMS plugin of Redmine.[37]

The code is fully managed using the GitLab[38] INAF repository,[39] including continuous integration (CI) at the subsystem level using the GitLab CI environment for automated subsystem verification. SonarQube[40] has been connected to the GitLab projects: the new code commit triggers the Sonar scanner, which provides the quality report and a tag pass/fail according to well-defined quality metrics. These tests are performed in a testing environment.

Docker containers[41] and an official ASTRI virtual machine are used for development, CI, and deployment.

5.3 Software Development Life Cycle

The software system engineering team has defined a software development plan that integrates aspects of Agile Development methodologies,[42] including (i) frequent iterations and releases; (ii) feature-driven development; (iii) unit and component tests created with the source code by the development teams during each iteration; (iv) automated testing and CI; and (v) distributed configuration management. The software system engineering team has also developed verification and validation plans. The quality assurance team defined the quality assurance plan for the software. All suppliers of the ASTRI software follow these plans. The supplier performs the verification procedures to test the system as a white box; the customer conducts the validation with the system as a black box to accept the delivered software.

The following major reviews are foreseen in the ASTRI Mini-Array software life cycle:

1. CoDR: this review demonstrates that a full view of the software complies with science requirements, system requirements, observing cycle, and operation concepts.
2. Preliminary design review (PDR): this review demonstrates that the preliminary design of the subsystem meets all system requirements with acceptable risk and within the cost and schedule constraints. It establishes the basis for proceeding with the detailed design. Documentation describing the baseline design is the output of this review. The end of this review starts the iterative and incremental phase of the development. Development and quality assurance plans were also delivered.
3. Critical design review (CDR): the scope of this milestone is to demonstrate that the design reached an appropriate level of detail to support the production of the code, AIV, and test, meeting all performance, scheduling, and operational requirements. This review is part of an iteration, but not all iterations foresee a formal CDR. A CDR is part of the iterative and incremental development approach, and the software is developed in parallel to this activity. Based on the iteration's scope, only documents could be updated to synchronize code and documents.
4. Acceptance test review (ATR): the scope of the review is to verify the completeness of the developed software, documentation, and test and analysis reports. Also, it ensures that the software reaches a level of maturity to be deployed. After this review, the software is delivered to the customer and deployed at the AOS or the data center.
5. Operational readiness review (ORR): the scope is to establish that the software system is ready for operations by examining test results, analyses, and operational demonstrations. It also shows that documentation is complete for each software configuration item. For SCADA, this review must be performed at the ASTRI AOS (the operational environment).

The project started with a general CoDR and some subsystem PDR reviews to provide a general decomposition of the project and a preliminary design. To integrate this first phase with a set of development iterations, a V-model which embedded the management of iterations and incremental deliveries has been adopted for the entire software life-cycle, as shown in Fig. 10; note that the adopted V-model does not imply that the development process is a waterfall method. In detail, the project started with the following phases:

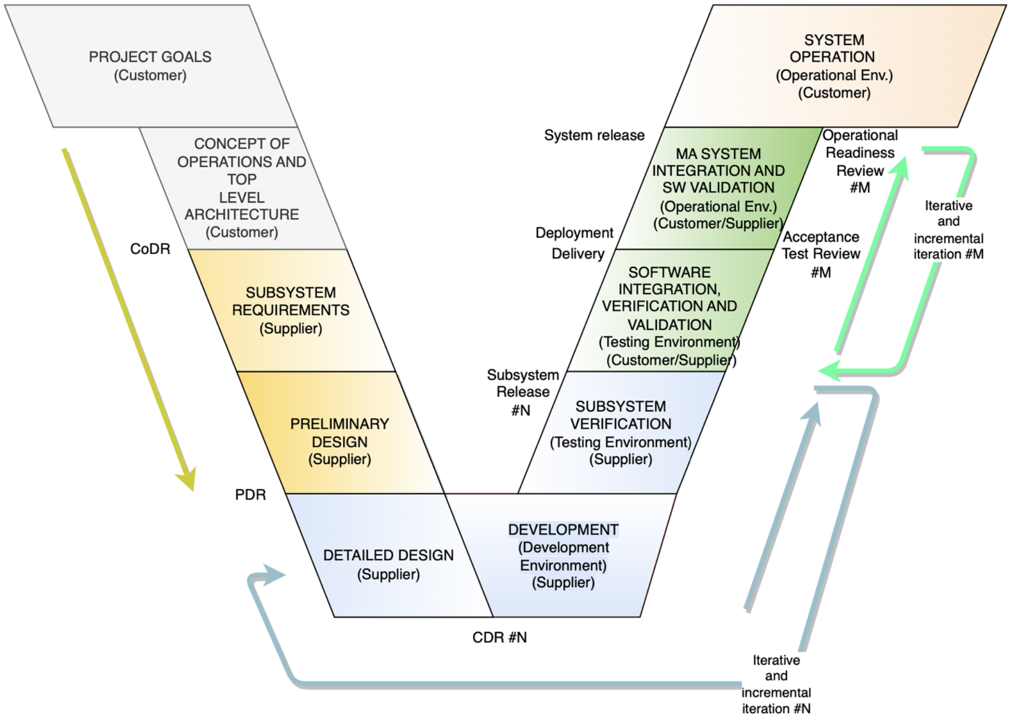
PROJECT GOALS (Customer)
CONCEPT OF OPERATIONS AND TOP LEVEL ARCHITECTURE (Customer)
CoDR
SUBSYSTEM REQUIREMENTS (Supplier)
PRELIMINARY DESIGN (Supplier)
PDR
DETAILED DESIGN (Supplier)
DEVELOPMENT (Development Environment) (Supplier)
CDR #N
SUBSYSTEM VERIFICATION (Testing Environment) (Supplier)
Subsystem Release #N
SOFTWARE INTEGRATION, VERIFICATION AND VALIDATION (Testing Environment) (Customer/Supplier)
Deployment Delivery
MA SYSTEM INTEGRATION AND SW VALIDATION (Operational Env.) (Customer/Supplier)
System release
SYSTEM OPERATION (Operational Env.) (Customer)
Operational Readiness Review #M
Acceptance Test Review #M
Iterative and incremental iteration #M
Iterative and incremental iteration #N

Fig. 10 ASTRI Mini-Array Software life-cycle and reviews. Gray boxes are part of the system definition phase. Yellow boxes are part of the subsystem requirement and preliminary design phase, closed by a PDR. Blue boxes represent a subsystem development iteration closed by a subsystem software release. All subsystem software releases are aligned by a milestone of a software system (e.g., SCADA). The first green box is the software system integration, verification, and validation phase, closed by an acceptance test review. The second green box is the system software deployment, including hardware assembly integration and integration with other software system integration (e.g., SCADA with the DPS), closed by an operational readiness review.

1. System definition phase (gray boxes in Fig. 10): the entire software system has been defined. This phase was closed in June 2020 by a CoDR conducted by a panel of external reviewers. After this review, the set of documents described in Sec. 4.3 was released. The PBS, part of these deliverables, was used to define the WBS of the software for the definition of the customer-supplier chain and assigning responsibility for each software subsystem.
2. Subsystem requirement and preliminary design phases (yellow boxes in Fig. 10). These phases are conducted at the subsystem level and is closed by a PDR. The main output is the detailed use cases and drafts of the software requirement document and the detailed design document. A risk analysis is also performed at this level. The only mandatory documents of this phase are the detailed use-case document and functional decomposition of the software. The set of documents and the level of details are agreed upon between the customer and the supplier. SCADA team conducted the PDR for some SCADA subsystems (telescope control system, monitoring system, array data acquisition system, and online observation quality system) in Spring 2021, with a panel of reviewers part of the software system engineering team.

At the end of each subsystem's PDR, the development starts iteratively and incrementally. The number and size of each iteration depend on the subsystem; iterations are agreed upon between customer and supplier and are based on the milestones foreseen by the ASTRI Mini-Array project connected with hardware procurement and related deployment. Each subsystem development iteration (blue boxes in Fig. 10) is divided in the following phases:

1. Detailed design: the starting point of each iteration is the selection of detailed use cases or only some steps of a detailed use case. The design or an update of the detailed design

documents released in a previous iteration is foreseen. The verification test plan is defined in advance for verification purposes for each iteration. The detailed design document is also updated at the end of the iteration before the release of the software.

2. Development of the software: software is developed, and documentation is updated. There are no constraints about the development methodologies adopted by each team; some teams use the Scrum methodology;[43]
3. Subsystem verification: all manual and automated verification tests are executed. For SCADA subsystems (e.g., telescope control system and monitoring system), the use of hardware simulator of assemblies that must be controlled or monitored is foreseen. The subsystem development iteration ends with the release of software and documents.

All subsystem releases are aligned with a software system (e.g., SCADA) milestone; the purpose of each milestone is defined at the system level. When all subsystems release the software for a specific milestone of a software system, the software integration, delivery, and deployment iteration starts and is divided into

1. Software integration, verification, and validation (first green box in Fig. 10): these steps allow the integration of all delivered subsystems of a software system (e.g., DPS or SCADA) in the representative testing environment. Verification procedures at system level are executed to demonstrate the success of integration. A preliminary software systems integration (e.g., SCADA with the DPS) is performed at this level. An ATR could be foreseen in executing validation procedures for major releases. At the end of this phase, the entire software system is delivered. The software system is ready to be deployed at the AOS (for SCADA) or in the data center.
2. Software validation and Mini-Array system integration (second green box in Fig. 10) allow the final deployment of a software system. SCADA is deployed at the AOS, while off-site software systems (e.g., DPS, science user support) are deployed in the data center. The archive system is distributed between off-site and on-site, but the final version of the archive system is off-site. The integration with the hardware assemblies of the mini-array system and related verification and validation procedures is foreseen at the AOS. The final software systems integration (e.g., SCADA with the DPS) is performed at this level. This phase is closed by an ORR for major releases.

At the end of this process, the software is used for system operations.

This process is not linear and sometimes requires some synchronization points between subsystems. After some iterations, we discovered that a general internal CDR for the SCADA subsystems developed by INAF was necessary. The primary purpose was to align the internal SCADA interfaces, verify the consistency of documentation and compliance with the top-level documents, align documents, including lessons learned during the iterations from other subsystems, and update the risk analysis after 1-year development. This review was conducted in the Spring of 2022 with the software system engineering team as the review panel.

In our approach, we have adopted the 12 Agile Methodology principles. These principles are highly useful for developing the ASTRI Mini-Array Software System, especially when dealing with the need to synchronize the development of many teams, and the deployment of integrated software with the on-site hardware. These principles are applied in this context in the following way:

1. Prioritize customer satisfaction: the customers of this project are the scientists. Scientific requirements of the ASTRI Mini-Array have been considered the main drivers since the beginning of the project.
2. Welcome changing requirements: in a project as complex as this, changes are inevitable, and not all hardware specifications are available at the beginning of the project. An Agile approach allows us to adapt to changes and include new specifications without significant disruptions.
3. Deliver working software frequently: frequent releases ensure that each increment of the software can be tested and integrated with the available on-site hardware, allowing for early identification of issues. Current planning foresees a minor software system

release every 2 months and a major one every 8 months, including the ORR for major releases.

4. Collaborate with stakeholders: regular collaboration with astronomers, scientists, and other stakeholders ensures the software aligns with system and scientific goals. Continuous feedback from stakeholders is collected in use cases and periodically updated.
5. Build projects around motivated individuals: encouraging self-organizing teams to make decisions and adapt to challenges is part of the ASTRI development plan and the adopted customer-supplier relationship.
6. Use face-to-face communication: although geographically distributed, each release is organized around a Kanban board at the software system level, and we use video conferencing and collaboration tools for regular face-to-face communication to update the Kanban board, enhance understanding, and help the alignment of the project.
7. Working software is the primary measure of progress: use working software as the primary indicator of progress, ensuring that it meets the project's needs. For each minor and major release, we provide software working at Teide with available on-site hardware.
8. Maintain a sustainable pace: regular face-to-face and additional working meeting allow for the long-term productivity of the team.
9. Strive for technical excellence: encourage best practices in software development to ensure the software is reliable and maintainable for the project's needs.
10. Keep it simple: simplify complex tasks into manageable components, making it easier to coordinate the deployment of an integrated system. This work was done at the project's beginning with the initial CoDR.
11. Self-organizing teams: the ASTRI development plan allows each team to self-organize and make decisions regarding their work, promoting flexibility and creativity in finding solutions.
12. Reflect and adjust regularly: conduct regular retrospectives to identify what is working and what needs improvement regarding synchronization, collaboration, and development procedures.

For overall planning, strict integration and collaboration with the system engineering team to adapt the schedule with the real on-site hardware telescope is part of these principles. This collaboration ensures that the software development aligns with the system engineering requirements and adjusts to any changes in the hardware deployment schedule, facilitating a smoother integration process. Given the geographically distributed nature of the project and the need for synchronization during on-site deployment, Agile principles that include frequent communication can ensure that each team remains aligned with the overarching project goals.

5.4 Testing Environment

The testing environment of the software comprises two test beds able to reproduce the ICT infrastructure at the AOS,[15] including hardware assemblies simulators. One test bed hosts the same version of the software installed at the Teide site or is used for verification activities before the deployment, while the other test bed runs the software under integration. With this infrastructure, we can emulate the on-site ICT infrastructure and install, run and verify the software before the on-site deployment, including all hardware and services. The test bed is based on a virtualization system (ProxMox[44]) to run virtual machines that emulate the on-site ICT, including all interconnection functions of his local area network (LAN) and the necessary network services: domain name server, network address translator (NAT), and routing.

The test bed supports manual verification and validation procedures and supports also automated tests performed using the GitLab CI framework based on Docker containers.

5.5 Release Management

The release management concerns the whole software development life cycle. As presented in the previous sections, we are providing many releases according to the project schedule. Any release, in addition to the implemented software, shall include the specific document version of the requirement specification, detailed design, verification test plan, verification test report, and the user manuals related to the latest developed features. A validation test plan and test reports are foreseen for the acceptance of the software. Eventually, the release document, which collects all the deliverables for a release, shall be published and used for personnel training.

5.6 Software Quality Assurance Approach

According to the ASTRI Mini-Array Product Assurance Plan,[45] we also released a software quality product assurance plan (SPAP) to establish the goals, the processes, and the responsibilities to implement the effective quality assurance functions for the ASTRI Mini-Array software. The SPAP provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the product assurance manager, the product assurance responsible for the software and all the actors involved to monitor and assess software development processes and products.

6 Conclusions

This paper outlines the software architecture and engineering approach used for the ASTRI Mini-Array Software System. Its primary function is to manage observing projects for the array, which includes using both the Cherenkov camera (for celestial γ-ray and cosmic-ray investigations) and the stellar intensity interferometry detectors. The system is responsible for various tasks, such as observation handling, array control and monitoring, data acquisition, archiving, data processing, and simulations. It also supports users conducting Cherenkov and intensity interferometry observations and provides scientific tools for exploiting observational data.

The development plan for the software implementation covers all the project phases, from construction to operations and dissemination. This document outlines the primary requirements and constraints influencing the software's definition. To achieve this, the architecture, various views, different aspects of the ASTRI Mini-Array software, and significant architectural decisions have been discussed in the text.

The ASTRI Mini-Array project is also being developed to pave the way to participation in CTAO. In this respect, the ASTRI Mini-Array may be considered a pathfinder of CTAO for INAF and the other international partners involved in the project. In particular, for the SST sub-array of CTAO, not only the telescopes' optomechanical structure will be pretty similar, but also the telescope control system, including the engineering HMI, will be in practice the same (or, at least, the one of the ASTRI Mini-Array will be, for the most part, reused). In addition, the two projects share several technological and conceptual similarities, which is also part of the innovative and collaborative nature of the field. Many of the authors of this paper are actively involved in both projects, contributing with their expertise to develop software for CTAO and ASTRI Mini-Array projects. They make use of standard tools and technologies, such as ACS and OPC-UA. This fosters knowledge exchange between the two collaborations.

On the other hand, there are some significant differences between CTAO and the ASTRI Mini-Array projects, which led to different choices in terms of the respective software architectures:

1. The ASTRI Mini-Array has a much lower data rate than CTAO, which enables us to acquire and store all the data from the Cherenkov cameras. This also allows us to apply stereoscopic event reconstruction post-facto after data acquisition. As a result, there is no need for an on-site stereo-event analogic trigger, simplifying the architecture and providing greater flexibility in scientific data exploitation. This means that we can always change the stereoscopic event reconstruction pattern later to optimize data reconstruction;
2. The ASTRI Mini-Array lacks a real-time data analysis system to detect transients related to its scientific objectives, due to the previous choice. Nonetheless, the system will be tuned to react to external alerts promptly, within a few minutes, for remarkable exceptional astronomical events;
3. Due to the aforementioned points, on-site dynamic scheduling is unnecessary for the ASTRI Mini-Array;

4. Although there is no on-site dynamic scheduling for the ASTRI Mini-Array, the decision regarding which SBs should be executed during the night could be made autonomously by the software based on current conditions or manually by the astronomers who supervise the operations. The list of SBs is optimized and scheduled off-site to define the short-term and long-term observation plan. The short-term observation plan is prepared in advance with a pre-ordered list of SBs based on scientific priorities, target visibility, NSB level, and environmental conditions;
5. If the operating conditions of the telescopes fall outside the pre-selected range, such as e.g., the minimum number of operating telescopes, environmental conditions, and atmosphere characterization, the pre-selected OB of the ASTRI Mini-Array is cancelled or stopped if it was already in progress. In this case, the next OB is selected to continue the observations.

Finally, the high network bandwidth between the data center in Italy and the Teide Observatory has provided many benefits for the ASTRI Mini-Array, positively impacting software architecture. The data are transferred off-site as soon as an SB is closed, allowing us to perform initial data reconstruction, data quality assessment, and a scientific quick-look within a few minutes after the completion of the SB. Moreover, moving the data processing and scientific quick-look to the off-site data center simplifies the on-site ICT and software architecture. A full and automated synchronization of the archive system between on-site and off-site is performed, increasing the system's reliability. On the other hand, the telescope control will be entirely managed on-site but without the need for on-site operators who can remotely operate.

The lessons learned from the ASTRI Mini-Array project may also be very valuable to the CTAO software development (and, of course, the other way around). As both projects are advancing, we foresee several opportunities for code reuse, optimization, and collaboration, with mutual benefits. In light of these connections, we acknowledge the importance of a continued dialogue between the ASTRI Mini-Array project and the CTAO software development efforts.

Managing the software life cycle (design, development, verification, integration, validation, delivery, and deployment) for the ASTRI Mini-Array project is challenging. The development involves different software suppliers (INAF, with its Institutes distributed at different locations in Italy, the University of Geneva in Switzerland, AC3E in Chile, and IAC in the Canary Islands), with a customer suppliers chain operating at different levels within the project. Our managing approach follows standard procedures based on the ECSS and is properly adapted to the Mini-Array case. An Agile, iterative, and incremental process is pursued, with selected use cases assumed as the baselines for each iteration. This approach allows us to manage the complexity of the geographically distributed organization and to effectively support the incremental development of the ASTRI Mini-Array system at the Teide Observatory.

Code and Data Availability

Data sharing is not applicable to this article, as no new data were created or analyzed.

The ASTRI Mini-Array software developed by the ASTRI collaboration and used during operations, with only a few exceptions regulated by industrial contracts, is primarily governed by the LGPL from the Free Software Foundation. The software will become open-source as soon as a future fully operational version is released. Code will be available at the following link https://www.ict.inaf.it/gitlab/astri/

Acknowledgments

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 explicitly assigned to the Italian National Institute of Astrophysics (INAF). We acknowledge the support of the Brazilian Funding Agency FAPESP (Grant No. 2013/10559-5), the South African Department of Science and Technology through Funding Agreement 0227/2014 for the South African Gamma-Ray Astronomy Program and the ANID-Basal Fund, Project FB0008 (AC3E). IAC is supported by the Spanish Ministry of Science and Innovation (MICIU). They are partially supported by H2020-ASTERICS, a project funded by the European Commission Framework Programme Horizon 2020 Research and Innovation action under

Grant Agreement No. 653477. The ASTRI project is becoming a reality thanks to Giovanni "Nanni" Bignami and Nicoló "Nichi" D'Amico, two outstanding scientists who, in their capability as INAF Presidents, provided continuous support and invaluable guidance. While Nanni was instrumental in starting 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, and Massimo Cappi, the Coordinator of the High Energy branch of INAF, have been also very supportive of our work. We are very grateful to all of them. Unfortunately, Nanni and Nichi passed away, but their vision still guides us. We are very grateful for the software architecture and development support to Ismam Abu, Alessandro Carosi, Luca Castaldini, Elena Fedorova, Federico Fiordoliva, Michele Mastropietro, Francesco Visconti, and Georgios Zacharis. We thank Joe Schwarz, one of the main authors of the present paper, who passed away just during the review process. Joe was crucial in defining the control software architecture for the CTAO and ASTRI projects. His guidance was essential in selecting ACS as the environment. Joe initiated various development activities in this regard. Despite facing severe health issues, until the end, he remained updated on the progress of different activities and actively participated in the ASTRI meetings. This article has gone through the internal ASTRI review process.

References

1. G. Pareschi, "The implementation of the ASTRI Mini-Array gamma-ray experiment at the Observatorio del Teide, Tenerife," *Proc. SPIE* **12182**, 121820J (2022).
2. S. Scuderi et al., "The ASTRI Mini-Array of Cherenkov telescopes at the observatorio del teide," *J. High Energy Astrophys.* **35**, 52–68 (2022).
3. S. Vercellone et al., "ASTRI Mini-Array core science at the Observatorio del Teide," *J. High Energy Astrophys.* **35**, 1–42 (2022).
4. J. Hinton and W. Hofmann, "Teraelectronvolt astronomy," *Annu. Rev. Astron. Astrophys.* **47**(1), 523–565 (2009).
5. G. Sottile et al., "ASTRI-Horn Cherenkov camera: improvements on the hardware and software components," *Proc. SPIE* **12188**, 1218830 (2022).
6. https://ohwr.org/projects/white-rabbit (accessed 8 January 2024).
7. H. Brown, *The Intensity Interferometer: Its Application to Astronomy/R. Hanbury Brown*, Taylor and Francis, London (1974).
8. D. Kieda, S. Swordy, and S. Wakely, "A high resolution method for measuring cosmic ray composition beyond 10 TeV," *Astropart. Phys.* **15**(3), 287–303 (2001).
9. G. Pareschi et al., "The ASTRI SST-2M prototype and mini-array for the Cherenkov Telescope Array (CTA)," *Proc. SPIE* **9906**, 99065T (2016).
10. A. Bulgarelli et al., "The Software Architecture and development approach for the ASTRI Mini-Array gamma-ray air-Cherenkov experiment at the Observatorio del Teide," *Proc. SPIE* **12189**, 121890D (2022).
11. L. Zampieri et al., "A Stellar Intensity Interferometry Instrument for the ASTRI Mini-Array telescopes," *Proc. SPIE* **12183**, 121830F (2022).
12. G. Bonanno et al., "Focal plane detector and front-end electronics of the stellar intensity interferometry instrument for the ASTRI Mini-Array telescopes," *Proc. SPIE* **12183**, 1218322 (2022).
13. S. Germani et al., "The pointing monitoring camera hardware and software systems for the ASTRI Mini-Array project," *Proc. SPIE* **12188**, 1218835 (2022).
14. D. Impiombato, "UVSiPM: a light auxiliary detector to measure the night sky background seen by the ASTRI Mini-Array Cherenkov telescopes at the Observatorio del Teide," *Proc. SPIE* **12191**, 121910X (2022).
15. F. Gianotti et al., "ASTRI Mini-Array on-site information and communication technology infrastructure," *Proc. SPIE* **12189**, 121891D (2022).
16. https://opcfoundation.org (accessed 8 January 2024).
17. https://modbus.org/ (accessed 8 January 2024).
18. N. Parmiggiani et al., "The online observation quality system software architecture for the ASTRI Mini-Array project," *Proc. SPIE* **12189**, 121892H (2022).
19. P. Kruchten, "The 4+1 view model of architecture," *IEEE Software* **12**(6), 42–50 (1995).
20. https://www.uml.org/ (accessed 8 January 2024).
21. A. Bulgarelli et al., "The Cherenkov Telescope Array Observatory: top level use cases," *Proc. SPIE* **9913**, 991331 (2016).
22. L. Macaulay et al., "USTM: a new approach to requirements specification," *Interact. Comput.* **2**(1), 92–118 (1990).
23. http://ac3e.usm.cl/ (accessed 8 January 2024).

24. F. Russo et al., "The telescope control system for the ASTRI Mini array of imaging atmospheric Cherenkov telescopes," *Proc. SPIE* **12189**, 121892I (2022).
25. M. Corpora et al., "Design and development of the supervisor software component for the ASTRI Mini-Array Cherenkov Camera," *Proc. SPIE* **12189**, 121891Z (2022).
26. V. Conforti et al., "The array data acquisition system software architecture of the ASTRI Mini-Array project," *Proc. SPIE* **12189**, 121890N (2022).
27. V. Pastore et al., "Array data acquisition system interface for online distribution of acquired data in the ASTRI Mini-Array project," *Proc. SPIE* **12189**, 1218924 (2022).
28. F. Incardona et al., "The monitoring logging and alarm system of the ASTRI Mini-Array gamma-ray air-Cherenkov experiment at the Observatorio del Teide," *Proc. SPIE* **12189**, 121891E (2022).
29. https://confluence.alma.cl/display/ICTACS/ACS+documentation (accessed 8 January 2024).
30. G. Chiozzi et al., "CORBA-based common software for the ALMA project," **4848**, 43–54 (2002).
31. I. Prandoni et al., "The Sardinia Radio Telescope: from a technological project to a radio observatory," *Astron. Astrophys.* **608**, A40 (2017).
32. I. Oya et al., "The array control and data acquisition system of the Cherenkov telescope array," in *17th Int. Conf. Accel. and Large Exp. Phys. Control Syst.*, WEMPR005 (2020).
33. S. Lombardi et al., "The data processing, simulation, and archive systems of the ASTRI Mini-Array project," *Proc. SPIE* **12189**, 121890P (2022).
34. S. Germani et al., "The Stereo Event Builder software system of the ASTRI Mini-Array project," *Proc. SPIE* **12189**, 121891R (2022).
35. https://ecss.nl/ (accessed 8 January 2024).
36. https://sparxsystems.com/ (accessed 8 January 2024).
37. https://www.redmine.org/ (accessed 8 January 2024).
38. https://gitlab.com/ (accessed 8 January 2024).
39. https://www.ict.inaf.it/gitlab/astri/ (accessed 8 January 2024).
40. https://www.sonarqube.org/ (accessed 8 January 2024).
41. https://www.docker.com/ (accessed 8 January 2024).
42. http://agilemanifesto.org/ (accessed 8 January 2024).
43. https://www.scrum.org/ (accessed 8 January 2024).
44. https://www.proxmox.com/ (accessed 8 January 2024).
45. N. La Palombara et al., "The product assurance programme of the ASTRI mini-array project," *Proc. SPIE* **12187**, 121871I (2022).

Andrea Bulgarelli has experience in system and software processes and knowledge of instrument development and simulation for X- and gamma-ray telescopes, such as AGILE, ASTRI Mini-Array, and CTA Observatory. He performed technological and scientific research on multi-wavelength and multi-messenger astrophysics. He gained experience in requirement engineering, interface management, and co-engineering activities with system engineers and scientists to define the requirements and architecture of big science astrophysical projects.

Biographies of the other authors are not available.