



**HAL**  
open science

## The first release of the Cherenkov Telescope Array Observatory array control and data acquisition software

I Oya, P Aubert, L Baroncelli, P Bolle, P Bruno, A Bulgarelli, S Caroff, L  
Castaldini, V Conforti, A Costa, et al.

### ► To cite this version:

I Oya, P Aubert, L Baroncelli, P Bolle, P Bruno, et al.. The first release of the Cherenkov Telescope Array Observatory array control and data acquisition software. SPIE Astronomical Telescopes + Instrumentation 2024, Jul 2024, Montréal, Canada. pp.131011D, 10.1117/12.3017568 . hal-04669465

**HAL Id: hal-04669465**

**<https://hal.science/hal-04669465v1>**

Submitted on 3 Sep 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The first release of the Cherenkov Telescope Array Observatory Array Control and Data Acquisition software

I. Oya<sup>a</sup>, P. Aubert<sup>b</sup>, L. Baroncelli<sup>c</sup>, P. Bolle<sup>d</sup>, P. Bruno<sup>e</sup>, A. Bulgarelli<sup>c</sup>, S. Caroff<sup>b</sup>,  
L. Castaldini<sup>c</sup>, V. Conforti<sup>c</sup>, A. Costa<sup>e</sup>, L. David<sup>f</sup>, G. De Cesare<sup>c</sup>, E. de Ona Wilhelmli<sup>d</sup>,  
A. Di Piano<sup>c</sup>, K. Egberts<sup>g</sup>, R. Fernandez<sup>h</sup>, V. Fioretti<sup>c</sup>, S. Fukami<sup>c</sup>, E. Garcia<sup>b</sup>, E. García<sup>b</sup>,  
H. Gasparyan<sup>d</sup>, S. Germani<sup>k</sup>, C. Hoischen<sup>g</sup>, F. Incardona<sup>e</sup>, D. Kostunin<sup>d</sup>, B. Lopez<sup>a</sup>, E. Lyard<sup>l</sup>,  
G. Maurin<sup>b</sup>, D. Melkumyan<sup>d</sup>, K. Munari<sup>e</sup>, T. Murach<sup>d</sup>, A. Muraczewski<sup>m</sup>, N. Nakhjiri<sup>j</sup>,  
D. Neise<sup>a, i</sup>, G. Panebianco<sup>c</sup>, N. Parmiggiani<sup>c</sup>, E. Pietriga<sup>f</sup>, V. Pollet<sup>b</sup>, I. Sadeh<sup>d</sup>, S. Sah<sup>h</sup>,  
A. Sarkar<sup>d</sup>, M. Schefer<sup>l</sup>, T. Schmidt<sup>d</sup>, D. Soldevila<sup>j</sup>, C. Steppa<sup>d</sup>, A. Tramacere<sup>l</sup>, R. Vallés<sup>j</sup>,  
T. Vuillaume<sup>b</sup>, and F. Werner<sup>n</sup>, for the CTA Observatory<sup>o</sup>

<sup>a</sup>Cherenkov Telescope Array Observatory gGmbH, Saupfercheckweg 1, 69117 Heidelberg,  
Germany.

<sup>b</sup>Univ. Savoie Mont Blanc, CNRS, Laboratoire d'Annecy de Physiques des Particules-IN2P3,  
74000 Annecy, France.

<sup>c</sup>INAF INAF/OAS Bologna, Via Gobetti 93/3, 40129 Bologna, Italy.

<sup>d</sup>Deutsches Elektronen-Synchrotron DESY, Platanenallee 6, 15738 Zeuthen, Germany.

<sup>e</sup>INAF, Osservatorio Astrofisico di Catania, Via S Sofia 78, I-95123 Catania, Italy.

<sup>f</sup>Université Paris-Saclay, CNRS, Inria, Gif-sur-Yvette, France.

<sup>g</sup>Institut für Physik & Astronomie, Universität Potsdam, Karl-Liebknecht-Strasse 24/25, 14476  
Potsdam, Germany.

<sup>h</sup>COSYLAB JSC, Control System Laboratory, Gerbičeva ulica 64, 1000 Ljubljana, Slovenia

<sup>i</sup>ETH Zürich, Institute for Particle Physics and Astrophysics, Otto-Stern-Weg 5, 8093 Zürich,  
Switzerland.

<sup>j</sup>Institute of Space Sciences(ICE,CSIC), and Institut d'Estudis Espacials de Catalunya(IEEC),  
Carrer de Can Magrans, s/n 08193 Cerdanyola del Vallés, Spain.

<sup>k</sup>Università di Perugia, Dipartimento di Fisica e Geologia, Italy.

<sup>l</sup>Département d'Astronomie, Université de Genève, Chemin d'Ecogia 16, CH-1290 Versoix,  
Switzerland.

<sup>m</sup>Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, ul. Bartycka 18,  
00-716 Warsaw, Poland.

<sup>n</sup>Max-Planck-Institut für Kernphysik, D 69029 Heidelberg, Germany.

<sup>o</sup><http://www.cta-observatory.org>

## ABSTRACT

The Cherenkov Telescope Array Observatory (CTAO) is the next-generation atmospheric Cherenkov gamma-ray project. CTAO will be deployed at two sites, one in the Northern and the other in the Southern Hemisphere, containing telescopes of three different sizes for covering different energy domains. The commissioning of the first CTAO Large-sized Telescope (LST-1) is being finalized at the CTAO Northern site. Additional calibration and environmental monitoring instruments such as laser imaging detection and ranging (LIDAR) instruments and weather stations will support the telescope operations. The Array Control and Data Acquisition (ACADA) system is the central element for onsite CTAO operations. ACADA controls, supervises, and handles the data generated by the telescopes and the auxiliary instruments. It will drive the efficient planning and execution of

observations while handling the several Gb/s camera data generated by each CTAO telescope. The ACADA system contains the CTAO Science Alert Generation Pipeline – a real-time data processing and analysis pipeline, dedicated to the automatic generation of science alert candidates as data are being acquired. These science alerts, together with external alerts arriving from other scientific instruments, will be managed by the Transients Handler (TH) component. The TH informs the Short-term Scheduler of ACADA about interesting science alerts, enabling the modification of ongoing observations at sub-minute timescales. The capacity for such fast reactions – together with the fast movement of CTAO telescopes – makes CTAO an excellent instrument for studying high-impact astronomical transient phenomena. The ACADA software is based on the Alma Common Software (ACS) framework, and written in C++, Java, Python, and Javascript. The first release of the ACADA software, ACADA REL1, was finalized in July 2023, and integrated after a testing campaign with the LST-1 finalized in October 2023. This contribution describes the design and status of the ACADA software system.

**Keywords:** Manuscript format, template, SPIE Proceedings, LaTeX

## 1. INTRODUCTION

The Cherenkov Telescope Array Observatory (CTAO) is an initiative to build the next-generation atmospheric Cherenkov gamma-ray observatory.<sup>1,2</sup> CTAO will consist of two facilities, one in the Southern Hemisphere (close to the Paranal Observatory, Chile) and the other in the Northern Hemisphere (at the Roque de Los Muchachos Observatory, La Palma, Spain). In total, more than 60 Cherenkov telescopes will be deployed in the two sites, with additional auxiliary instruments, constituting one of the largest astronomical installations under development.<sup>2</sup>

The Cherenkov telescopes and additional auxiliary instruments – collectively known as array elements (AEs) – will be controlled and supervised by operators using the Array Control and Data Acquisition (ACADA) system from control rooms at the sites. ACADA is also responsible for handling, compressing, and storing the acquired data on disk. ACADA includes a pipeline that takes a quick look at the acquired data for early identification of transient phenomena or issues in the acquired data – the Science Alert Generation Pipeline (SAG), see Sec. 3.5.

Data acquired by ACADA are archived and processed by the Data Processing and Preservation System (DPPS), which will be hosted in two onsite data centres next to the telescopes (where ACADA also runs) and four offsite data centres located in Europe. The Science User Support System (SUSS), which will be managed from the CTAO Science Data Management Centre in Zeuthen, Germany, will make processed data and science tools available to end-users. SUSS will also provide the interface for scientists to create and submit CTAO observation proposals, and will generate the long- and mid-term observation schedules. In addition, the SUSS team is responsible for providing the Science Analysis Tools (SAT) to researchers using CTAO data. The SAT are based on the Gammapy open-source Python package.<sup>3</sup>

The ACADA collaboration was formally established between 2019 and 2020, but most of the partners forming the ACADA collaboration had been prototyping and collaborating as part of the CTAO Consortium activities during earlier years.<sup>4,5</sup> Thanks to this work, the requirements, architecture, design, and development process of ACADA were quickly formalized and finalized and underwent a successful preliminary design review (PDR) in 2020. This followed two major releases of the software, followed by a successful integration and test (I&T) campaign with the first CTAO telescope, the Large-Sized Telescope Prototype (LST1)<sup>6</sup> located in La Palma. In January 2024, the ACADA WP underwent a critical design review (CDR). With the design and development process of ACADA validated by the reviews and successful campaigns, the ACADA team is working on the next versions of the product to support the Observatory development schedule and the integration of the array elements, the science verification activities, and eventually early science activities.

This contribution describes the design and status of the ACADA software system. After introducing the driving requirements of ACADA, an overview of the ACADA architecture and technology stack is provided, followed by a brief description of the development process in ACADA. Then the capabilities provided by ACADA Release 1 (REL1) are presented. The main technical challenges and expected scope of the next versions of ACADA are then described. A summary of the contribution and status of the ACADA system concludes the paper.

## 2. DRIVING REQUIREMENTS FOR THE ACADA SYSTEM

ACADA is a central system in observatory operations, and as such the product requirements of ACADA (level-B requirements in the CTAO jargon) were amongst the first ones to be worked out in the CTAO. Almost 300 ACADA level-B requirements were identified, specified, and validated as a joint effort of the CTAO Systems Engineering, Science Operations, and Computing Department teams, together with the ACADA experts. These requirements have been further validated via requirements review, PDR, and critical design review (CDR) events, and are under configuration control. They have been traced down to ACADA subsystem level requirements (level-C requirements in the CTAO jargon) and connected to ACADA functions, use cases, components, and interfaces following a model-based systems engineering approach (MBSE).<sup>7</sup> A description of some of the important ACADA capabilities prescribed by the requirements is given in the following list:

- Operation of at least eight independently operable groups of telescopes (subarrays) at the same time, dynamically formed from distinct subsets of available telescopes at a given site.
- Support up to 99 telescopes in addition to light detection and ranging (LIDAR) instruments, stellar photometers, all-sky cameras, weather stations, ceilometers, and external light sources for the calibration Cherenkov telescope cameras (*illuminators* in CTAO terminology).
- Collect event data from all CTAO telescope cameras. The data volume to handle are: 24 Gb/s for each LST (up to four telescopes in each site), 12 Gb/s for each MST (up to 25 telescopes in CTAO-S and 19 in CTAO-N), and 2 Gb/s for each SST (up to 70 telescopes in CTA-S).
- Reduce the data volume of the Cherenkov telescope camera on the fly, such that the total data rate delivered to the DPPS at each site is less than 5 Gb/s while minimizing the impact on science. (A factor up to 50 once the whole set of telescopes is deployed).
- Acquire all hardware-related monitoring points provided by the AEs, up to an overall maximum of 200,000 data points.
- Handle errors and alarms of ACADA and AEs, suggesting to the CTAO operator possible mitigation actions to be taken.
- Interface international networks for astrophysical transients; receiving, processing and filtering incoming information, triggering observations within 5 seconds of the receipt of the external alert.
- Provide preliminary data reduction and analysis and conduct a full field-of-view search for transient and/or time-variable phenomena from those data.
- Automatic generation, and subsequent revision of the observational schedule by ACADA during an observing night without the need for human intervention, but with operators informed of the current schedule at all times.
- Automatic operation throughout the observing night in the absence of alarms.
- Capability for a full restart of ACADA software within 2 minutes, with full monitoring and alarm functionality of all available AEs available within 5 minutes.
- Suited to be operated by two single individuals: an operator and one support astronomer located in a control room.
- The core functionality of ACADA, needed for system control and the safe execution of observations, is to be available during 99% of the time in which observations are possible.
- Be deployable in any standard computing node running a Red Hat Linux derivative in the onsite data centre (the current valid version being AlmaLinux 9\*), and does not require specially tailored machines.

---

\*<https://almalinux.org>

### 3. ACADA SYSTEM ARCHITECTURE

The ACADA architecture design, derived following MBSE techniques<sup>7</sup> seeks to address the ACADA requirements while facilitating the development of its functions in well-decoupled subsystems and components allocated to the various team members of the geographically distributed ACADA collaboration. ACADA is designed as a highly reliable and fault-tolerant system, which will enable CTAO to operate simultaneously several subarrays, react on timescales of seconds to incoming external and internal science alerts, coordinate observation between the two array sites, and promptly detect malfunctioning instruments.

ACADA is a central element in each of the CTAO sites (CTAO-N and CTAO-S) and interfaces with many other CTAO systems. As illustrated in Fig. 1, ACADA interfaces with the following CTAO systems: the telescopes, the Array Calibration System, the Environmental Monitoring System, the DPPS, the Science Operations System (SOSS), the SUSS, the integrated protection systems (IPS), infrastructure elements such as the Onsite Data Centre and the power management system, and the Technical Operations Support System (TOSS). Describing the systems surrounding ACADA is outside the scope of this work, but details on the CTAO system architecture can be found in ref.<sup>8</sup> Additionally, ACADA has direct interfaces with external collaborating scientific facilities (for scheduling coordination and handling external science alerts), and the laser traffic control system at each of the sites (for coordination of the laser shots at each site).

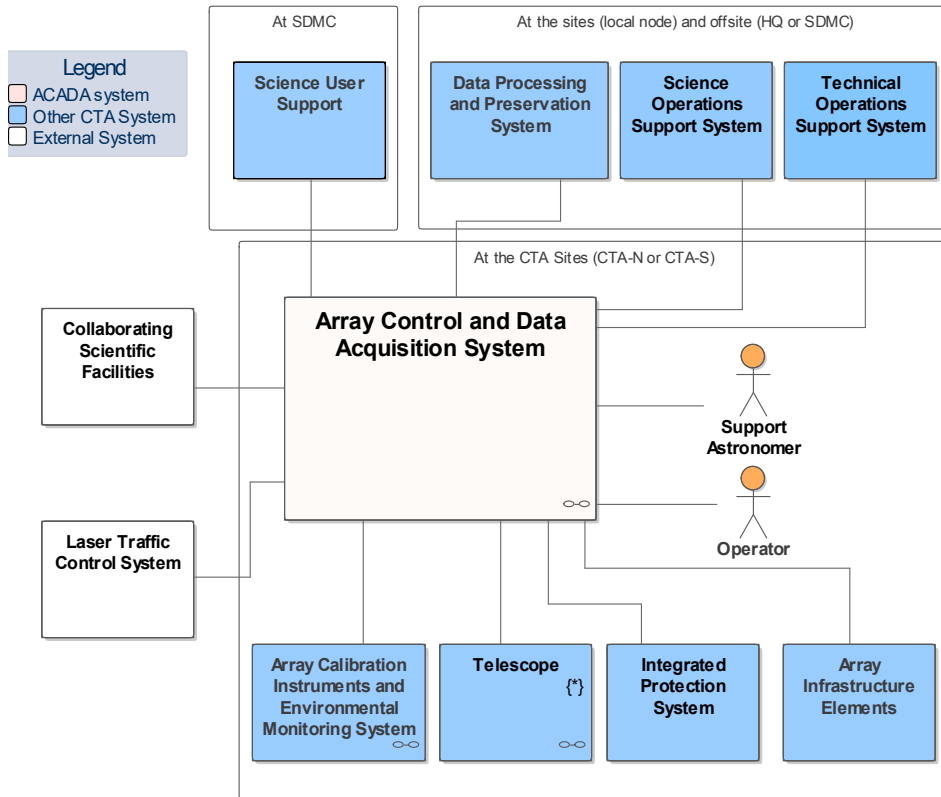


Figure 1. ACADA system context. The systems external to ACADA are depicted in blue (when part of the CTAO System) and in white (when external to the CTAO). The stick figures represent the human actors interacting with the system.

ACADA is a distributed system that uses the Alma Common Software (ACS) framework<sup>9</sup> as its basic middleware. ACADA provides its overall functionality via components, where each component is responsible for one or a few clearly defined tasks. The majority of the ACADA components are implemented as ACS components. They run on standard computing nodes of the onsite data centre at the site. The various ACADA components

interact to provide the functionality required of the ACADA as a whole. ACADA relies on ACS for base functionality towards that interaction and uses ACS for component instantiation and other such services. Each ACADA component has a supervisor that will monitor its existence and integrity, and orchestrate replacement of that component with its successor as needed. Component supervision and replacement are organized according to a supervision tree hierarchy.<sup>10</sup> ACADA includes the following items in its technology stack:

- **Middleware:** ACS<sup>b</sup>, OPC UA<sup>c</sup> (for access to monitoring data only)
- **Programming languages:** Python, Java, C++, Javascript (for the Human-Machine Interface (HMI) front-end only)
- **Databases:** MySQL<sup>d</sup>, MongoDB<sup>e</sup>, Redis<sup>f</sup>, Cassandra<sup>g</sup>
- **Messaging and serialization:** Apache Kafka<sup>h</sup>, ZeroMQ<sup>i</sup>, Google Protocol Buffers<sup>j</sup>
- **Workload management system:** Slurm<sup>k</sup>
- **Containers:** Docker<sup>l</sup>
- **Other:** ESO's Integrated Alarm System(IAS)<sup>m</sup>

The subsystems of ACADA are illustrated in Fig. 2 and described in the remainder of this section.

### 3.1 Resource Manager

The Resource Manager (RM) is the subsystem responsible for the supervision of all other systems of ACADA, and any external system under its supervision such as the control systems of the AEs, as well as for coordinating the allocation of telescopes to subarrays. The RM includes the root node of the supervision tree. Melkumyan et al.<sup>10</sup> provide more details on the RM.

### 3.2 Central Control

The Central Control (CC) subsystem is responsible for the execution of the scheduling blocks (SBs) provided by the short-term scheduler by sending corresponding commands to the telescopes and other controllable AEs, while supervising the ongoing operations, and overseeing the Array Data Handler (ADH). Melkumyan et al.<sup>10</sup> provide more details on the CC.

### 3.3 Human-Machine Interface

The HMI offers a comprehensive view of the status of the observations to the operators and support astronomers located in the control room of the CTAO installations, providing means to supervise and interact with the array elements, check the status of the environment, insert SBs, and monitor their execution.<sup>11</sup>

---

<sup>b</sup><https://confluence.alma.cl/display/ICTACS/>

<sup>c</sup><https://opcfoundation.org>

<sup>d</sup><https://www.mysql.com>

<sup>e</sup><https://www.mongodb.com>

<sup>f</sup><https://redis.io>

<sup>g</sup>[https://cassandra.apache.org/\\_/index.html](https://cassandra.apache.org/_/index.html)

<sup>h</sup><https://kafka.apache.org>

<sup>i</sup><https://zeromq.org>

<sup>j</sup><https://protobuf.dev>

<sup>k</sup><https://slurm.schedmd.com>

<sup>l</sup><https://www.docker.com>

<sup>m</sup><https://integratedalarmsystem-group.github.io>

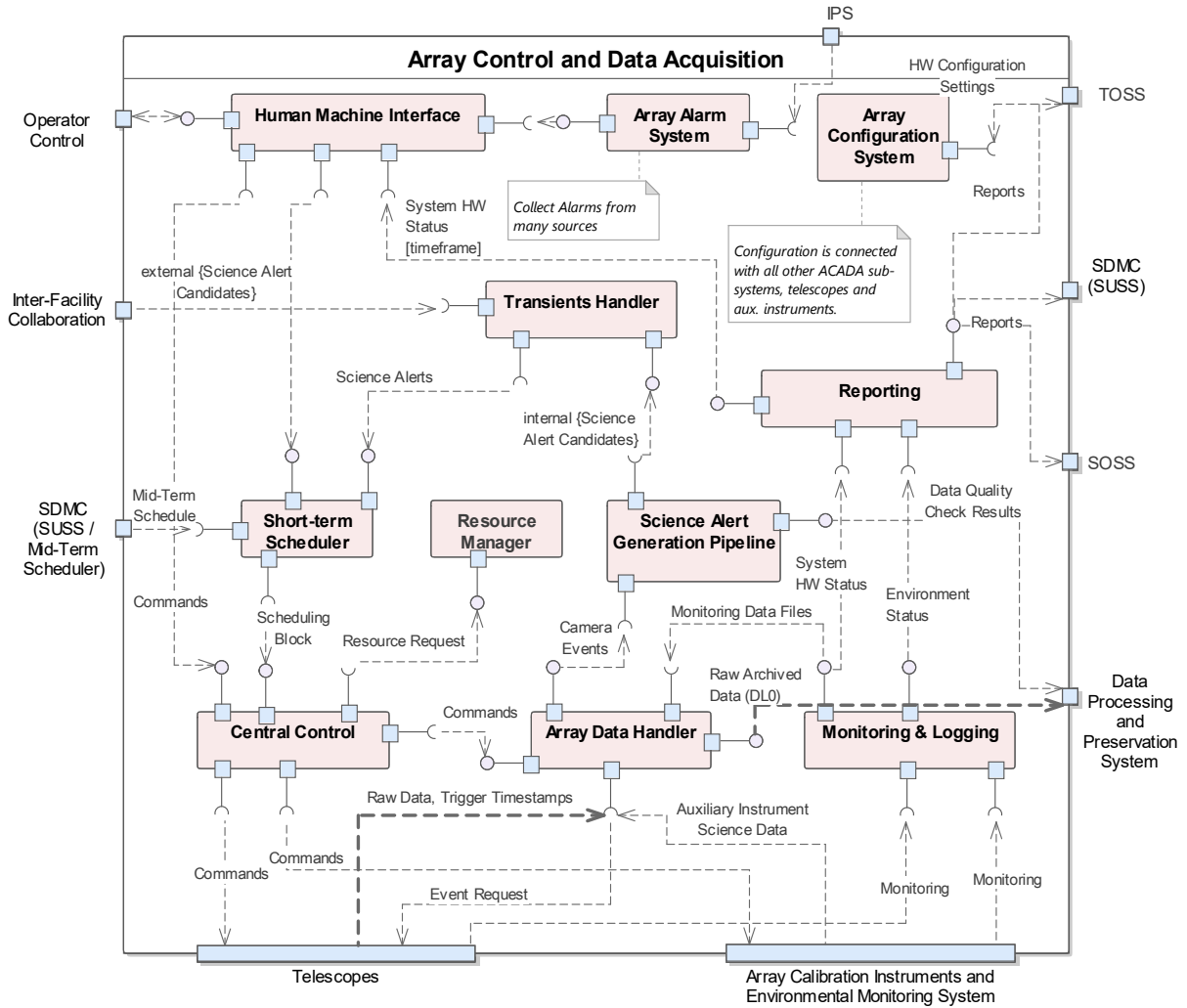


Figure 2. Logical view of ACADA, representing the main components of the system. Only the highest-level components and the most relevant data elements and interfaces are shown. The diagram uses the UML component notation, and the basic entities are the software components, depicted as pink boxes. Blue squares are “ports” that identify the interfaces of the system. Those ports in the boundary of the ACADA identify the external interfaces of the system and those in the components identify the internal interfaces. The dashed lines show the flow of data elements and the arrows the direction of the flow. The thicker data flow line indicates the flow of Cherenkov camera data through ACADA. The ACADA Command Line Interface and the ACADA Task Manager are not shown in this diagram.

### 3.4 Array Data Handler

The ADH subsystem is responsible for handling the stream of data from the array instrumentation.<sup>12</sup> Includes components to handle the Cherenkov camera data and to reduce the received volume of data and provides the array-level trigger. The same infrastructure is used for handling acquired data by auxiliary instrumentation onboard telescopes, as well as for the Array Calibration and Environmental Monitoring System devices. It provides local temporary storage of data until the onsite DPSS system receives that data, eventually transferring it offsite.

### 3.5 Science Alert Generation Pipeline

The SAG is a pipeline running online that performs a quick look analysis of the acquired data and produces data quality indicators, which are exposed to the support astronomer at the control room and transferred to the DPPS. It also serves to generate internal candidate scientific alerts, submitted to the Transients Handler (TH). SAG also performs science monitoring (generation of sky maps and light curves) for the support astronomer located in the control room. SAG uses Gammapy<sup>3</sup> in the SAG pipeline. Bulgarelli et al.<sup>13</sup> provide more details on the SAG.

### 3.6 Short-Term Scheduler

The Short-Term Scheduler (STS) subsystem is responsible for deciding, in real-time, how to group and use the telescopes of a CTAO installation to perform nightly operations, based on a mid-term Schedule supplied to it before the beginning of the night operations. This sub-system is also responsible for reacting in real-time to changing environmental conditions and to science alert observation requests by the TH.

### 3.7 Transients Handler

The TH subsystem responsible for managing internal and external transient science alert candidates by filtering, processing, and ranking them, submitting SBs to the STS, and requesting immediate reaction if needed. See Steppa et al.<sup>14</sup> for more details.

### 3.8 Monitoring and Logging Systems

The Monitoring and Logging Systems (collectively known as MON) are components providing services for monitoring and logging AE and ACADA internal data. The monitoring components acquire data items from the telescopes and other devices deployed at the CTAO array sites and make those data immediately available for the HMI and the SAG quality checks. The monitoring data are stored for later detailed inspection and are made available to DPPS and TOSS. MON includes the capabilities for acquiring and storing logging data of AEs and ACADA, and a log analyser to help identifying issues and support preventive maintenance activities. See Costa et al.<sup>15</sup> for more details.

### 3.9 Array Alarm System

The Array Alarm System (AAS) is a subsystem that provides the service that gathers, filters, exposes, and persists all the relevant alarms raised by both the ACADA processes and AEs under the supervision of the ACADA system. It is capable of identifying new alarms based on the analysis and correlation of the system software logs acquired by the MON. The data from the alarms are sent to the operator via the HMI. AAS uses the IAS.<sup>16</sup> See Costa et al.<sup>15</sup> for more details.

### 3.10 Array Configuration System

Array Configuration System, also known as the ACADA configuration database (CDB), stores and distributes the configuration datasets of the ACADA system components and the supervised AE systems. The datasets include the software deployment configuration and the settings of the instrumentation.

### 3.11 Reporting System

The Reporting System (REP) is responsible for gathering the relevant data from the other ACADA subsystems in order to produce status and quality reports of the ACADA operations for the HMI and for other systems outside the ACADA, in particular SOSS and TOSS. It also includes the operator shift log tool.

### 3.12 Other system elements

ACADA also provides supporting libraries and applications to facilitate the operation of the ACADA subsystems, as well as to provide finer-grained interfacing with the system to support testing, commissioning, and troubleshooting. Most prominently, ACADA incorporates a command line interface (CLI) that allows for direct interaction with the ACADA subsystems.

A *Task Manager* functionality is the component responsible for triggering and executing ACADA-level tasks, such as handling acquired data to DPPS, at certain times every day.



## 4. DEVELOPMENT PROCESS

The ACADA team organization was presented in ref.<sup>17</sup> and the approach to testing and quality assurance (QA) in ref.<sup>18</sup> The ACADA product will be delivered as a set of eight major releases with increasing capabilities, where intermediate, finer-grained versions will also be made available.<sup>19</sup> The overall development process is incremental and iterative, with each major release expected to be done in timescales from 6 months to one year. Inside each major release, the team utilizes a *scrum-like* approach with time-boxed sprints of three weeks with a defined list of features (normally use cases) to be delivered, tested, and integrated into the ACADA master branch.

The ACADA QA process is mainly based on the execution of tests in continuous integration (CI) pipelines<sup>20</sup> (via Jenkins), additional integration test executed manually,<sup>21</sup> and static code analysis displayed via Sonarqube.<sup>18</sup> For every major ACADA release, an increasingly more stringent set of QA metrics is planned, as well as a program for testing with AEs, being simulators, real instrumentation from AEs such as cameras in test stands in laboratories or factory locations, or AEs already deployed in the CTAO sites.

The first version of ACADA, Release 0 (REL0) was announced in 2021 and focused on assembling the ACADA systems for the first time based on existing prototype versions of ACADA subsystems and adding them to the CI pipelines. The first use-case-based end-to-end tests were implemented and executed as well, with the main goal of exercising the ACADA software development life-cycle SDLC<sup>20</sup> as a team for the first time. Once REL0 was finalized, a retrospective workshop was carried out to gather feedback from the extended ACADA team, which was followed by adjustments to the SDLC to suit the team's preferences. REL0 was only deployed in the ACADA test environment.<sup>20,21</sup>

## 5. RELEASE 1 CAPABILITIES

ACADA REL1 was released in July 2023 and focuses on functional requirements and single telescope operations. It aims to be fully capable of operating an individual CTAO Large-sized Telescope. The core functionality provided is the following:

- Full control capabilities of the LST1 via the ACADA CC
- Basic version of the supervision tree in place (only Java and Python implemented in this release)
- Manage the system startup
- Implement the data handling of the LST1 according to the defined raw and archival data models
- Central trigger with events unique identifiers assignment for a single telescope events
- HMI, to be deployed in the control room of the technical building, focused on SB submission and execution status, and system startup and shutdown representation.
- Storage of monitoring and logging data from LST1 assemblies and ACADA internal processes
- Manual scheduling support, queuing, validating, and submitting SBs added via the HMI and the TH (no on-line optimization)
- Handle gamma-ray burst science alerts, which will be observed as soon as it is possible (either immediately, or later in the night)
- Simple Hillas analysis<sup>22</sup> with the production of an event list, and basic data quality checks.<sup>13</sup>

In order to realize the prescribed goals, ACADA REL1 incorporates the following characteristics:

- 79 ACADA level-B requirements and 34 use cases implemented and formally verified
- More than 250 level-C requirements implemented and formally verified

- Continuous integration pipeline up and running
- 290k lines of code, with a 62% average line code coverage
- Quality indicators as described in Tab. 1.

Table 1. Quality gates for ACADA REL1.

Metric	Target
Maintainability rating	B <sup>(ii)</sup> (technical debt ratio max. 10%)
Reliability rating	C <sup>(ii)</sup> (no blocker and no critical bugs)
Security hotspots	100% reviewed <sup>(ii)</sup>
Duplicated lines	<3% <sup>(i, ii)</sup>
Line coverage	>50% <sup>(ii, iii)</sup>
Automated tests	100% passed <sup>(iii)</sup>
Verification by inspection, demonstration, or analysis	100% passed

Notes: (i) For ACADA REL1 the *duplicated lines* metric should not be applied to the HMI sub-system, because it is believed that this metric might add only limited value for front-end HMI code, since this usually makes use of repetitive patterns, (ii) determined by SonarQube, (iii) obtained via tests executed by Jenkins.

The ACADA software was first tested in the ACADA CI environment, then in the CTAO test cluster using a telescope simulator provided by the LST Consortium, and then with the real telescope during fall 2023.<sup>21</sup> All the objectives of the testing campaigns were achieved, and detected issues were documented, and are being addressed as part of the work the the next versions of ACADA.

With the finalization of ACADA REL1, an estimated 50% of the total ACADA code has been delivered, covering all core functionalities. In terms of ACADA use cases, those realizing the core business of ACADA are already implemented, however, many alternative and exception paths are not yet verified. Non-core use cases have been postponed to future releases. In terms of formal verification of ACADA requirements, 25% are verified. Since the focus of REL1 was to provide the ACADA core functionality, most of the non-functional requirements, in particular performance and scalability requirements, were planned for future ACADA releases.

After REL1 was made available, an I&T campaign was executed by the ACADA and LST1 telescope teams during fall 2023. During these I&T activities, ACADA and its interfaces to telescopes were extensively tested for the first time in realistic conditions, with the main goal of retiring risks such as having inadequate design solutions.<sup>21</sup> A consequence of the I&T campaign was the discovery of some performance and interface issues. Most internal ACADA product issues were addressed via bug-fix releases during the I&T campaign or within the next few weeks after. Performance optimizations and the resolution of instabilities have been added to the scope of the current intermediate ACADA release. Moreover, the need for some adjustments in the application programming interface (API) that governs the ACADA-telescope interface and in the structure of the raw camera data generated by the telescopes was identified.

## 6. SCOPE OF NEXT VERSIONS OF ACADA

The objectives of the next two major releases of ACADA, Release 2 (REL2) and Release 3 (REL3), focus on scalability, performance, and autonomy. In particular, they will support the following:

- **Multi-telescope support:** ACADA will support the operation, data acquisition, and analysis of data from four telescopes of the same type on REL2, and five telescopes in up to two subarrays in REL3. The three types of ACADA telescopes will be supported.
- **Other array elements:** ACADA will support operations with the stellar photometers and illuminators in REL2, and LIDARs in REL3.

- **Automatic operations:** Starting in REL2 and with enhancements in REL3, ACADA will be capable of executing observation without human intervention during the night thanks to the capabilities of the STS. An increasing range of science cases will be supported by the TH, which will cooperate with the STS to execute those observations.
- **HMI panels:** Further HMI panels will be deployed, including panels to display monitoring and status of the health of the AEs, with enhanced usability.
- **Performance:** ACADA restart times, time to react to changes in the schedule on the fly, and other timing-related performance requirements will be optimized to reach the few minutes timescales.
- **Availability:** The supervision tree of ACADA will be fully implemented, allowing for a fast recovery of failing components and processes.
- **Technical debt:** The technical debt will be reduced by addressing issues, improving the performance, and required adjustments on interfaces following the findings in REL1 I&T campaigns.

Later ACADA releases (Release 4 and beyond) will focus on other aspects such as the operation of more telescopes, array elements, and support of more concurrent subarrays. Additionally, further features such as more sophisticated handling of alarms, and automatic generation of night key performance indicators (KPIs) will be supported. As described in Sec. 4, with every ACADA release more stringent QA goals will be defined. For REL2, the QA objectives are described in Tab. 2.

Table 2. Quality objectives for ACADA REL2.

Metric	Target
Maintainability rating	A (technical debt ratio max. 5%)
Reliability rating	C (no blocker and no critical bugs)
Security rating	A (no vulnerabilities)
Security hotspots	100% reviewed
Duplicated lines	<3% (<5% for the HMI)
Line coverage	>60%
Automated tests	100% passed
Verification by inspection, demonstration, or analysis	100% passed
Automated tests related to requirements covered in previous releases	100% (no regressions)

## 7. SUMMARY AND OUTLOOK

After years of design and prototyping work, the ACADA team released a version of the software capable of operating a real CTAO telescope. 50% of the ACADA codebase has been implemented, with a good fraction of the ACADA use cases and requirements verified and more than 60% of the code covered by tests. Successful tests with the LST1 and a favourable CDR validate the maturity of the software and provide valuable input for improvements.

Further capabilities will be incrementally added in the following ACADA software releases, allowing the operation of CTAO configurations and achieving the objectives of the CTAO construction plan, as the first telescopes are accepted in the next couple of years.

## ACKNOWLEDGMENTS

The authors acknowledge the support and help of all ACADA team members, as well as CTAO Computing, Systems Engineering, and Project Management department members in establishing the management structure of ACADA. The authors acknowledge the fundamental commitment to ACADA, with personnel and equipment, by Deutsches Elektronen-Synchrotron (DESY), Institut de Ciències de l'Espai (ICE/CSIC), Istituto Nazionale di Astrofisica (INAF), University of Geneva – Astronomy Department, Laboratoire d'Annecy de Physique des Particules (LAPP), Nicolaus Copernicus Astronomical Center of the Polish Academy of Sciences (CAMK), Max-Planck-Institut für Kernphysik (MPIK) and the University of Potsdam.

The authors acknowledge the excellent support from the LST Consortium which permitted the ACADA team to test the ACADA REL1 with the LST1. The authors also acknowledge the help and support by the team responsible of the computing cluster located in La Palma owned by the University of Tokyo where the ACADA REL1 was deployed for the tests with the LST1. Both the possibility of using the LST1 and the computer cluster was fundamental for the execution of the ACADA test program.

The authors gratefully acknowledge financial support from the agencies and organizations listed here: <https://www.ctao.org/for-scientists/library/acknowledgments/>

## REFERENCES

- [1] The CTA Consortium, [*Science with the Cherenkov Telescope Array*], World Scientific Publishing Co. Pte. Ltd (2019).
- [2] McMuldloch, S. et al., “The Cherenkov Telescope Array Observatory (CTAO): ongoing development of the world’s premier ground-based gamma ray observatory,” in [*Proc. SPIE*], **13094**, 19 (2024).
- [3] Donath, A. et al., “Gammapy: A python package for gamma-ray astronomy,” *A&A* **678**, A157 (2023).
- [4] Oya, I. et al., “Designing and prototyping the control system for the Cherenkov Telescope Array,” in [*ACAT 2017 Procs.*], Contribution 42 (2017).
- [5] Oya, I. et al., “The Array Control and Data Acquisition System of the Cherenkov Telescope Array,” in [*7th Int. Conf. on Acc. and Large Exp. Physics Control Systems*], WEMPR005 (2019).
- [6] Abe, H. et al., “Observations of the Crab Nebula and Pulsar with the Large-sized Telescope Prototype of the Cherenkov Telescope Array,” *ApJ* **956**, 80 (2023).
- [7] Oya, I. et al., “The Software Architecture to Control the Cherenkov Telescope Array,” in [*Proc. SPIE*], **9913**, 991303 (2016).
- [8] Oya, I. et al., “A system architecture approach for the Cherenkov Telescope Array,” in [*Proc. SPIE*], **10705**, 107050Y (2018).
- [9] Chiozzi, G. et al., “Corba-based Common Software for the ALMA project,” in [*Proc. SPIE*], **4848**, 43 (2002).
- [10] Melkumyan, D. et al., “The Resource Manager and Central Control Systems for the Cherenkov Telescope Array Observatory,” in [*Proc. SPIE*], **13101**, 93 (2024).
- [11] Sadeh, I. et al., “The Graphical User Interface of the Operator of the Cherenkov Telescope Array,” in [*Proc. ICALEPCS 2017*], **User Interfaces and User eXperience (UX)**, TUBPL06 (2017).
- [12] Lyard, E. et al., “End-to-end data acquisition pipeline for the Cherenkov Telescope Array,” in [*35th International Cosmic Ray Conference, Proceedings of Science*], **301**, 843 (2017).
- [13] Bulgarelli, A. et al., “The Science Alert Generation system of the Cherenkov Telescope Array Observatory,” in [*Proc. SPIE*], **13101**, 84 (2024).
- [14] Steppa, C. et al., “Updates on the Transients Handler of the Cherenkov Telescope Array Observatory,” in [*Proc. SPIE*], **13101**, 65 (2024).
- [15] Costa, A. et al., “The comprehensive monitoring, logging, and alarm system for the Cherenkov Telescope Array Observatory,” in [*Proc. SPIE*], **13101**, 114 (2024).
- [16] Caproni, A. and Schmid, E., “The Integrated Alarm System for the ALMA Observatory,” in [*Proc. ICALEPCS 2017*], **Integrating Diverse Systems**, MOCPL07 (2017).
- [17] Oya, I., “Supporting the coordination of a software work-package of the Cherenkov Telescope Array via model-driven methodologies,” in [*Proc. SPIE*], **11450**, 1145011 (2020).

- [18] Antolini, E. and Oya, I., “Quality Assurance Plan for the Scada System of the Cherenkov Telescope Array Observatory,” in [*17th Int. Conf. on Acc. and Large Exp. Physics Control Systems Procs.*], Contribution MOMPL001 (2019).
- [19] Conforti, V. et al., “The Software Version Control Procedure for the Array Control and Data Acquisition Software of the Cherenkov Telescope Array Observatory,” in [*Proc. SPIE*], **13101**, 82 (2024).
- [20] Murach, T. et al., “Software testing for the cta observation execution system,” in [*Proc. SPIE*], **10707**, 107070D (2018).
- [21] Lopez, B. et al., “Methodology for the integration of the Array Control and Data Acquisition System with array elements of the Cherenkov Telescope Array Observatory,” in [*Proc. SPIE*], **13101**, 16 (2024).
- [22] Hillas, M., “Cerenkov light images of EAS produced by primary gamma rays and by nuclei,” in [*Proc. 19th ICRC, La Jolla*], **3**, 445 (1985).