Astronomy and Computing 57 (2026) 101171

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Architecture and design of the ESO ELT control software

Gianluca Chiozzi *, Luigi Andolfato, Javier Argomedo, Nicolas Beneš, Carlos Diaz Cano, Robert Frahm, Josef Hofer, Bogdan Jeram, Mario Kiekebusch, Nick Kornweibel, Ulrich Lampater, Federico Pellegrin, Andres Ramirez, Marcus Schilling, Babak Sedghi, Heiko Sommer, Arturo Hoffstadt Urrutia

European Southern Observatory, Garching bei, München, Germany

ARTICLE INFO

Keywords:
ELT
Telescope control system
Telescope control software
Architecture
Design

ABSTRACT

The Extremely Large Telescope (ELT) is a 39-meter optical telescope under advanced construction in the Chilean Atacama desert. The optical design is based on a five-mirror scheme and incorporates adaptive optics. The primary mirror consists of 798 segments. First scientific light is planned for the end of 2030. The status of the project is described in (Tamai et al., 2026).

The major challenges for the control of the telescope and the instruments are in the number of sensors (∼25000) and actuators (∼20000) to be controlled in a coordinated fashion, the computing performance and low latency requirements for the phasing of the primary mirror, performing adaptive optics, and coordinating all subsystems in the optical path.

Industrial contractors are responsible for the low-level control of individual subsystems and ESO for the development of coordination functions and control strategies requiring astronomical domain knowledge.

In this paper we focus on the architecture and the design of the control software in the context of the control system as a whole.

1. Introduction

The ELT is the largest and most complex ground-based telescope under construction, having a large segmented primary mirror with a diameter of 39m and a novel 5-mirror optical design.

To provide the science instruments in the two Nasmyth foci with a wavefront having an error in the nm range, the control software must carefully orchestrate the operation of the ELT's five mirrors (Fig. 1). In particular,

- The 798 M1 segments can be moved individually in piston, tip and tilt to control the shape of the mirror, keeping the relative edge displacements within a few nm. This is achieved via the position actuator (PACT) and edge sensor (ES) devices. The PACTs can move the segments to an absolute position and measure their placement. The ESs measure the relative displacement in piston, shear, and gap between adjacent segments. Position adjustments are computed and applied in real-time by a control loop (see (Reinacher et al., 2024) for details on the control algorithm) made of three stages: one stage for reading the absolute and relative measurements, one for computing the new target positions and one for sending the corrections to the actuators. The stages are pipelined and executed at 500Hz each. In addition, warping harness devices (WH), made of nine motors integrated in each segment support, are used, during telescope pointing, to compensate for gravity, temperature shifts, and manufacturing imperfections by applying forces to the segments.
- M2 and M3 are respectively convex and concave 4m mirrors. The mirror cells provide positioning capability for realigning the mirror within the telescope, and shape adjustment capability to compensate for constant errors.
- The deformable M4, with a diameter of 2.4m, can adjust its shape with 5352 actuators at rates up to 1 kHz (Biasi et al., 2016) to perform Adaptive Optics (AO), which mainly focuses on compensating for the effect of atmospheric turbulence. AO has a temporal and spatial bandwidth of the corrections of up to hundred Hz and about 50 cycles over the pupil diameter with a sampling rate up to 1 kHz. When bright enough sky targets are available, they feed natural guide star wavefront sensors inside the telescope Pre-Focal Station

* Corresponding author.
E-mail address: gchiozzi@eso.org (G. Chiozzi).

https://doi.org/10.1016/j.ascom.2026.101171
Received 5 May 2025; Accepted 21 June 2026
Available online 22 July 2026

(PFS) and/or inside the instruments. Their measurements drive the real-time corrections sent to M4. When sky targets are too faint, as it is often the case, 6 lasers mounted onto the telescope will generate artificial guide stars, by exciting the sodium atoms found in a layer of the atmosphere at 90km altitude. This sodium light is measured by laser guide star wavefront sensors, exclusively found inside the instruments, driving similar correction commands sent to M4.

• The M5 is a flat mirror that includes a fast tip-tilt for image stabilization for frequencies up to a few Hz. Its purpose is to compensate for perturbations caused by wind, atmospheric turbulence, and the deformations of the telescope itself.

Given the limited stroke (100 nm) of the M4 actuators and the limited capture range of the wavefront sensors in the guide probes, all mirrors are adjusted cooperatively to redistribute the slowly building non-zero-mean components in low order modes of M4 to the other degrees of freedom.

Focusing is performed by axially translating the M2 (major, occasional re-setting), the M4 hexapod (large amplitudes, low bandwidth) or applying a small curvature term on the otherwise flat M4 (continuous, fine adjustments).

Realignment -or rigid body motion- is performed by re-centering specific mirror units (e.g. the M2 unit).

Figure control, i.e. dynamically maintaining alignment and shape of the mirrors, applies to the individual M1 segments, the M2, M3 and M4.

Each of the two Nasmyth platforms A and B accommodate multiple instruments, which interface with the telescope through a Pre-Focal Station (respectively PFS-A and PFS-B) — the final opto-mechanical element in the optical path before the light reaches the focal plane. Each PFS has three sensor arms with Shack-Hartmann wavefront sensors and imaging cameras used to measure pointing errors, image motion and optical aberrations.The PFS serves two purposes: first, it relays the light collected by the telescope to the science instruments and to the Phasing and Diagnostic Station (PDS); second, it picks off and conditions the light from the three guide stars for use in acquisition, guiding, and wavefront sensing..

The Phasing and Diagnostic Station provides the sensors to bring the shape and position of the ELT mirrors from integration tolerances to deliver diffraction-limited performance in the ELT focal planes during AIV commissioning phase of the ELT. It also provides the sensoric means to phase the primary mirror, close high order AO loops, and diagnose the telescope on optical aspects throughout its lifetime during regular operations.

The instruments require high performance in terms of wavefront quality to reach their underlying scientific goals. This calls for higher order wavefront control than what can be delivered by the telescope itself. For this reason, post-focal AO modules will be developed to feed different science instruments.

The ELT Control Software has the purpose of monitoring, controlling, and coordinating with high precision timing all these devices (plus many others, like the telescope dome), distributed inside the dome, in an area that is almost 100m in diameter.

In this paper, we will first provide some necessary context information on the main characteristics and challenges of the ELT control system and of the corresponding control software; we will then describe the architecture and some aspects of the design of the main components. A description of the operation workflow will put the architectural choices in context.

We do not present details on development process, development environments, user interfaces, operating system. For these, look at (Pellegrin and Rosenquist, 2017) and (Hoffstadt et al., 2024).

The conclusions will provide information on the status of the project.

2. The ELT control system

The ELT Control System includes computers, network connectivity and software required to control all the subsystems of the telescope (Fig. 2), down to but not including the sensors and actuators. It also contains the High-Level Coordination and Control software (HLCC), the Wave Front Real-time Computer (WFRTC), and the Engineering Archive (EA) for all engineering data collected during the lifetime of the observatory.

Use of the control system is not limited to science operations of the commissioned telescope: it is first used in the Assembly Integration and Verification (AIV) phases of the ELT construction, as a support to commissioning and verification. After AIV, as maintenance activities are defined and implemented, the control system supports daytime activities, monitors the telescope and ensures safety. As a calibration tool the control system supports execution of defined calibration sequences.

Given the unprecedented scale and control complexity of the project, it is planned that requirements will evolve over the construction period, including AIV and commissioning, as operational experience allows us to refine how the components are orchestrated to reach the required performance.

The main requirements and challenges can be summarized as follows:

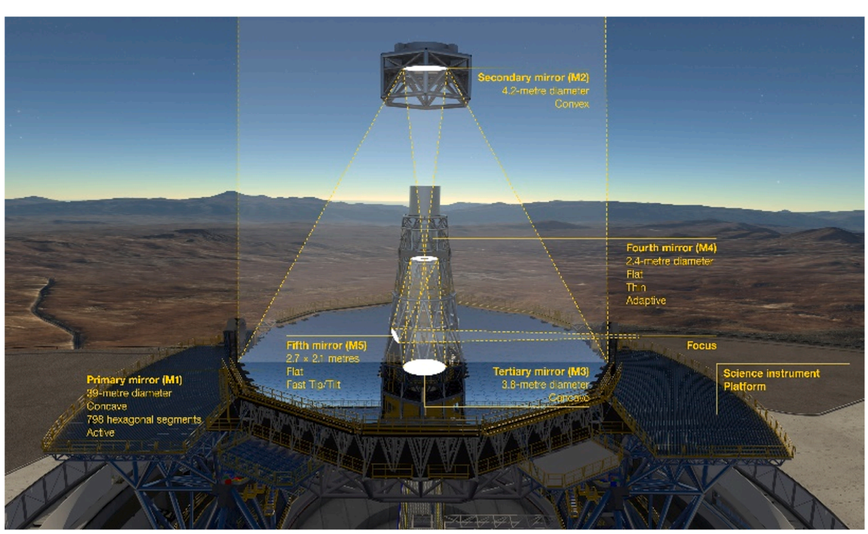

Fig. 1. Telescope subsystems following the light path.

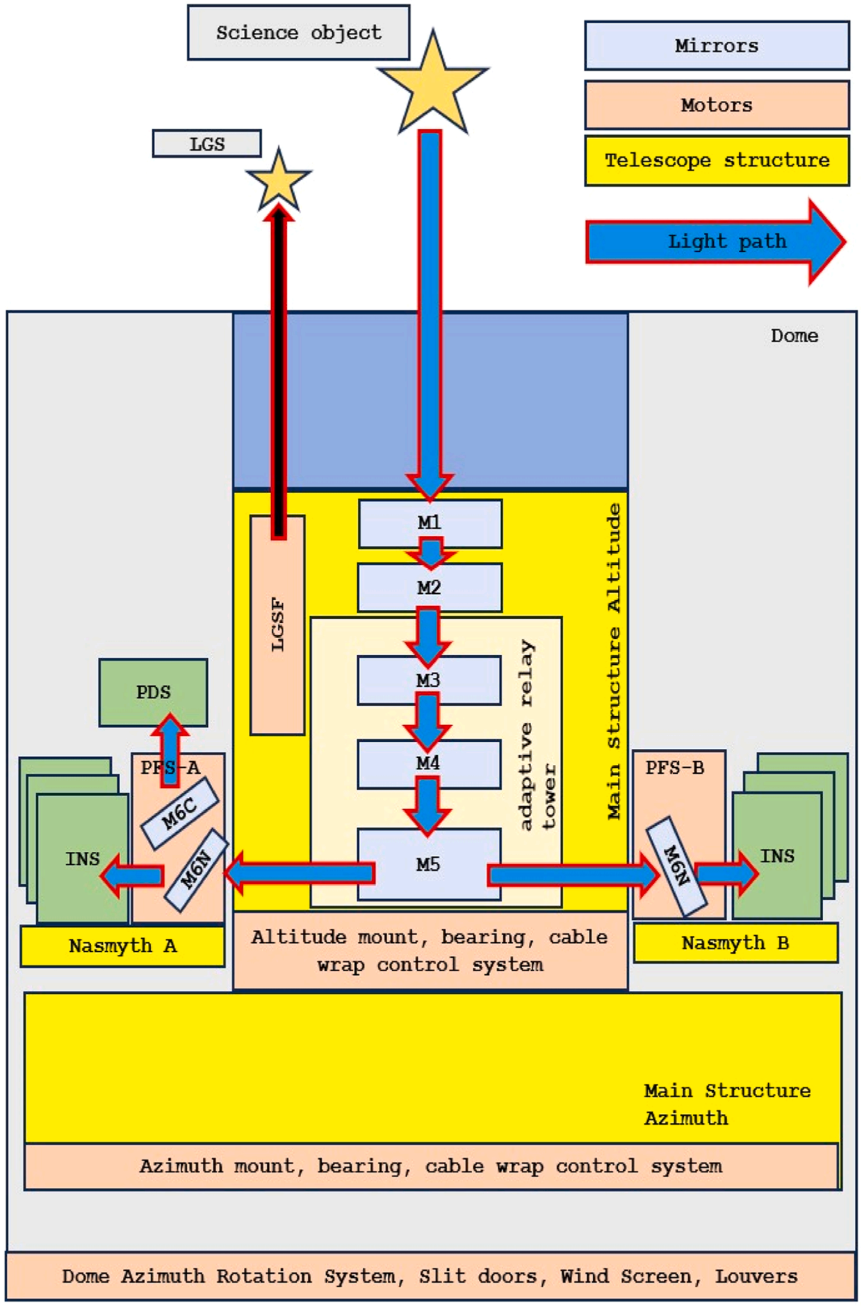

Fig. 2. Telescope subsystems following the light path.

- Size: ∼10000 tons of steel and glass to control, ∼20000 actuators, ∼1000 mirrors.
- Number of monitor and control points: ∼60000 I/O points.
- Number of interfaces: 12 subsystems, 10 focal stations, interfaces with instruments, interfaces with observatory services (for example site monitor and laser traffic control) and interfaces with operators.
- Number of instruments: 6 currently planned and 2 AO modules.
- High computational and data throughput requirements: ∼700 Gflops of computing performance and ∼17 GB/s of real-time data handling in AO.
- Heterogenous control loop frequences: from 0.01Hz to kHz rates.
- Microsecond-level synchronization: Across distributed control layers and subsystems..
- Software-intensive distributed control strategy.

Programmatic and organizational challenges are also important to consider.

The ELT Telescope procurement strategy foresees the outsourcing of all components and services which can be efficiently performed by industrial partners from ESO member states, while maintaining in-house those tasks for which ESO has specific domain expertise.

For the ELT Control System this same procurement principle applies. Thus, the overall system is constituted of components designed, built and delivered by industrial partners distributed across Europe, as well as by in-house projects. This geographic and organizational distribution of the development immediately enforces the need for clear identification of components and interfaces which should match not only a functional breakdown of the control system, but also reflect the organizational boundaries of the many development centers.

Instruments are instead developed by consortia from universities or astronomical research institutes, with technical support from ESO.

3. Architecture

The organizational and procurement requirements described above drive the architectural breakdown of the control system.

Fig. 3 shows an overview of the ELT CS, with some simplifications and omissions for readability (*numbers in orange are used to identify items referenced as "Fig. 3-(#)" in the text below*).

- Each telescope subsystem, individually contracted to industry, is associated with its own independent Local Control System (LCS) (Fig. 3-(1)). Each LCS contains all the required control software, unit (s), devices and local communication infrastructure to monitor, command and safely operate the whole subsystem.
- The Central Control System (CCS) (Fig. 3-(2)) consists of computers, middleware and application software to integrate and coordinate the LCSs. It includes also an external CCS-INS interface (Fig. 3-(3)) to operate the telescope and to communicate with the instruments. Moreover, it includes the real-time computer required to coordinate and interface with all wavefront control functions, on which the Telescope Real Time Executor (TREx) (Fig. 3-(4) runs.
- CCS includes the Global Interlock and Safety system (ILS) as well, responsible for telescope-level safety (Fig. 3-(5)).
- Instruments (Fig. 3-(6)) are individually developed by consortia of ESO partner institutes. Each instrument includes an independent Instrument Control System (ICS) developed following ELT standards and using the ICS Framework (FW) (Kiekebusch, 2024) provided by ESO. Interfacing with the telescope is done exclusively through the CCS-INS interface (Fig. 3-(3)) using the functionality provided by the ICS Framework.
- The Time Reference System and the Networking Infrastructure are common services provided by the Control System to LCSs, CCS and ICSs.

We have identified the following main LCSs:

Dome: Control of dome rotation, slit doors, windscreen, louvers, thermal regulation, power distribution, building management system, fluids provisioning, handling devices and access control.

Main Structure (MS): Control of the main azimuth and elevation axis (with position and velocity control loops, brakes and clamps), cable wraps, M5 repositioning when changing observing focus, and other auxiliary devices.

M1: Control of Position Actuators, M1 Edge Sensors and warping harnesses, in situ or temporarily integrated in qualification test beds. It includes control of the electrical power and cooling distribution in the M1 cell.

M2: Control of hexapod and warping harness. It implements adjustment capabilities for re-aligning within the telescope and shape adjustment capability, as required by the wavefront control strategy.

M3: Control of hexapod and warping harness. It implements adjustment capabilities for re-aligning within the telescope and shape adjustment capability, as required by the wavefront control strategy.

M4: Control of mirror shape (5352 actuators), Nasmyth switching mechanism, mirror positioning hexapod and gas cooling.

M5: Control of fast tip/tilt steering functions, as well as a slow active alignment capability in piston, tip and tilt with an eccentric shaft with roller bearings.

PFS: Control of the two Pre-Focal Stations respectively on the

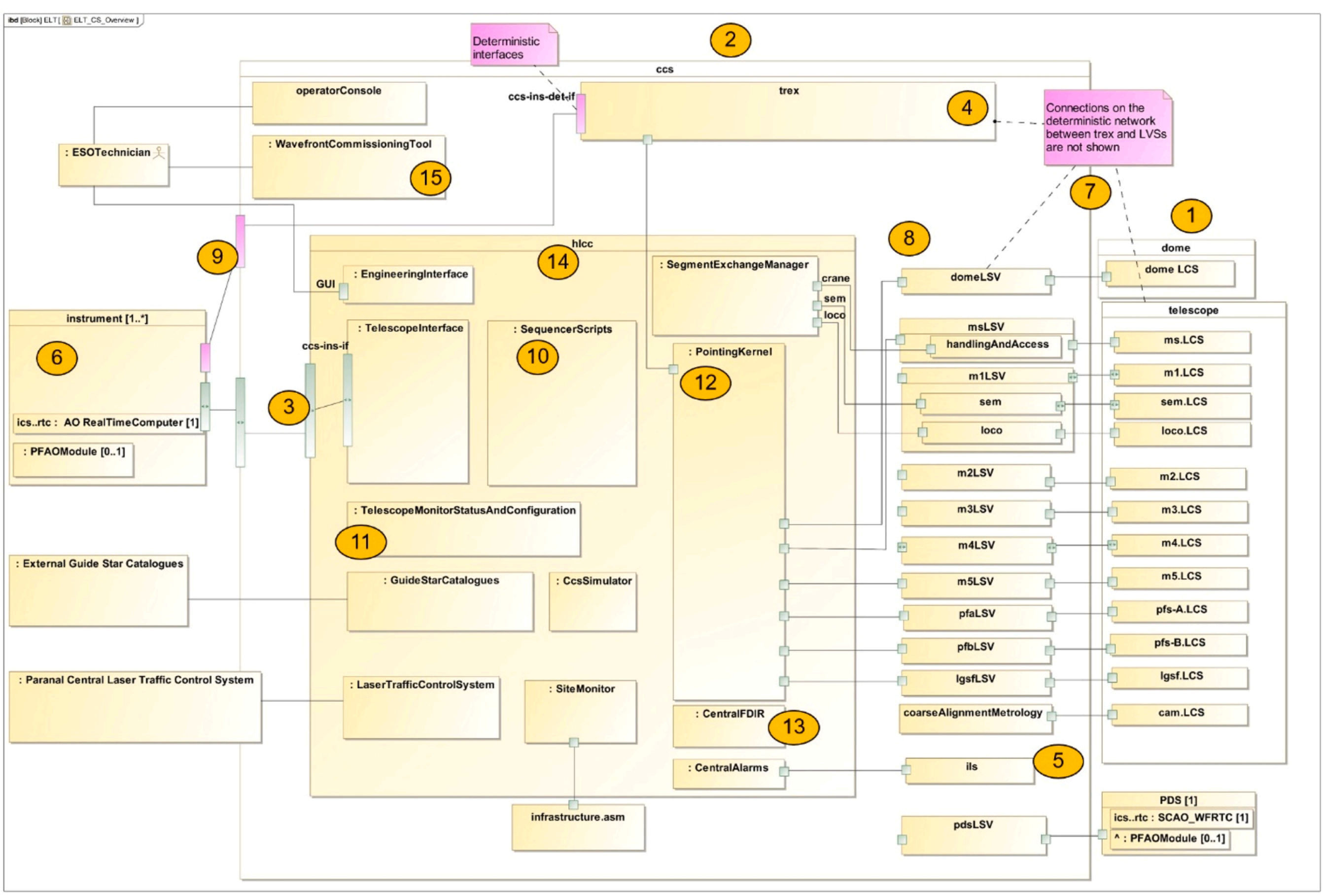

Fig. 3. ELT control system overview.

Narmyth A and B sides. It includes control of the three sensor arms' azimuthal and radial motion, field curvature compensation, focusing, camera pupil centering, calibration unit, shutter and filter wheel as well as the M6 mechanism to propagate the beam to the different instruments or to the Phasing and Diagnostic Station (PDS).

Laser Guide Star Facility (LGSF): Control of the Laser Guide Star Units.

Segment Manipulator (sem) and Local Coherencer (loco): Control of the devices respectively used to remove and insert segments during daily segment exchange operations and to phase the segments when they are installed.

Coarse Alignment Metrology (CAM): Control of the system to initially orient and maintain the positions of mirrors and optical systems. It provides the sub-millimeter positioning required before fine-phasing (wavefront sensing) can safely take over.

The LCS-CCS differentiation not only separates unit-level safety and control from telescope-level safety and control, but it also matches organizational boundaries in-line with the ELT procurement strategy:

- Individual subsystems (mirror units, main structure, dome, lasers) are designed, built and delivered by external industrial partners, responsible for the barebone control of the elementary functions of the devices.
- The integration of the subsystems to form the telescope system is carried out by ESO, leveraging our specific astronomical expertise.

The Network Infrastructure is split into dedicated virtual LANs to separate the traffic into:

1. Deterministic: High-performance distributed control loops (also across subsystems) are implemented using the MUDPI protocol (see below). Some examples are the AO loop and the M1 control loop.
2. Non-deterministic: the Data Distribution Service (DDS) is used for non-deterministic publisher/subscriber; ZPB (ZeroMQ + Protocol Buffers) for command/reply communication in the supervisory layers of the control system and therefore between applications running on Linux workstations. In most cases, OPC Unified Architecture (OPCUA) protocol is used for commands and measurements in the LCS-CCS communication. Non-deterministic monitoring data can also be published at low data rate on the Online Data Base (OLDB), a hierarchical key-value store/database, that provides a tree representation of the status and configuration of the whole telescope and of the instruments. Applications can subscribe to this data or do an explicit read access. This representation is particularly useful for data display and for exploring the system as a unified entity.
3. Precision Time Protocol: distribution of the observatory time and synchronization.
4. Management interface: system's management, software deployment and configuration.

MUDPI is a UDP data format developed specifically for all distributed control loops in the ELT (camera pixel streams, mirror and mount commands and measurements). It is a very basic multicast protocol, permitting non-intrusive interface verification by sniffing data on a separate host with a hardware timestamp-capable network card. This is a key approach to system and subsystem verification. The simplicity of MUDPI, a key requirement, makes it suitable for use in Ethernet-based distributed control loops and high-performance interfaces of the control system across languages and architectures (ESO).

Control of field devices from the LCSs foresees the use of two major Ethernet flavors of field bus: PROFINET IO and EtherCAT.

CCS interfaces with LCSs strictly via data communication over Ethernet. The software interfaces are defined in a series of Interface Control Documents (ICDs), Fig. 3-(7), specifying the logical addresses, data types, formats, rates and characteristics of data communication (Chiozzi et al., 2018). The LCS-CCS ICD specifications also separate control from safety.

In most cases, commands and measurements in the LCS-CCS communication use the OPC Unified Architecture (OPCUA) protocol. On the other hand, the non-deterministic interfaces between CCS components and between CCS and instruments typically are based on ZPB for commanding and DDS (plus Online Data Base) for monitoring while the deterministic interfaces rely on MUDPI.

The LCSs, CCS, and instruments interface with the observatory infrastructure for power, networking and access to the observatory clock.

4. Local control systems

The block diagram in Fig. 4 shows the structure of an LCS as a component of the associated subsystem.

An LCS contains two key functional groups: control and safety. Control functions enable standard operation of the subsystem, while safety functions preserve its integrity and guarantee the safety of personnel and equipment.

The control functions include Control Software, Local Control Unit (s), remote IO modules and a local communication infrastructure.

The safety functions include Safety Logic, Local Safety Unit(s), Safety IO devices and a fail-safe communication infrastructure.

The functions provided by the LCS can make no assumptions as to the nature of the subsystem use in the context of the telescope control system (the operation and wave front control strategies being implemented by the CCS, external to the LCS). For example, the M4 adaptive mirror must be available and operable to full performance irrespective of whether or not the telescope is observing or calibrating or parked.

The LCS provides interfaces to CCS that enable individual control of all subsystem devices and functions, irrespective of intention or mode, to the limits of the safety system. All subsystem functions must be controllable individually and independently. For example, it must be possible to move the warping harness of an M1 segment irrespective of the status of the edge sensors and position actuators.

LCSs share common requirements, such as:

- Continuous assessment of health and performance of the associated item.
- Detection, isolation and notification of faults and failures.
- Remote initialization, so that physical access or local intervention with equipment are not required.

All LCSs use as much as possible well-established industrial standards and Commercial Off-The-Shelf (COTS) products: custom-built solutions are strongly discouraged. Additionally, they are integrated with general-purpose connections to the field devices through the various Service Connection Points at the Telescope.

Local Control Units (LCUs) typically range from industrial Programmable Logic Controllers (PLCs) to general-purpose Linux-based workstations, depending on the specific performance requirements.

For two LCSs (the M1 and M4) it has been necessary to introduce bespoke, custom electronics:

- The M1 has custom sensors and actuators, with custom-built interface cards.
- The M4 incorporates FPGA and a proprietary field bus. Due to the high computation load, large actuator count and low latency requirements, combined with constraints on heat emissions, it was always planned that the M4 LCS would follow this development path, and as such it was one of the earliest contracts launched.

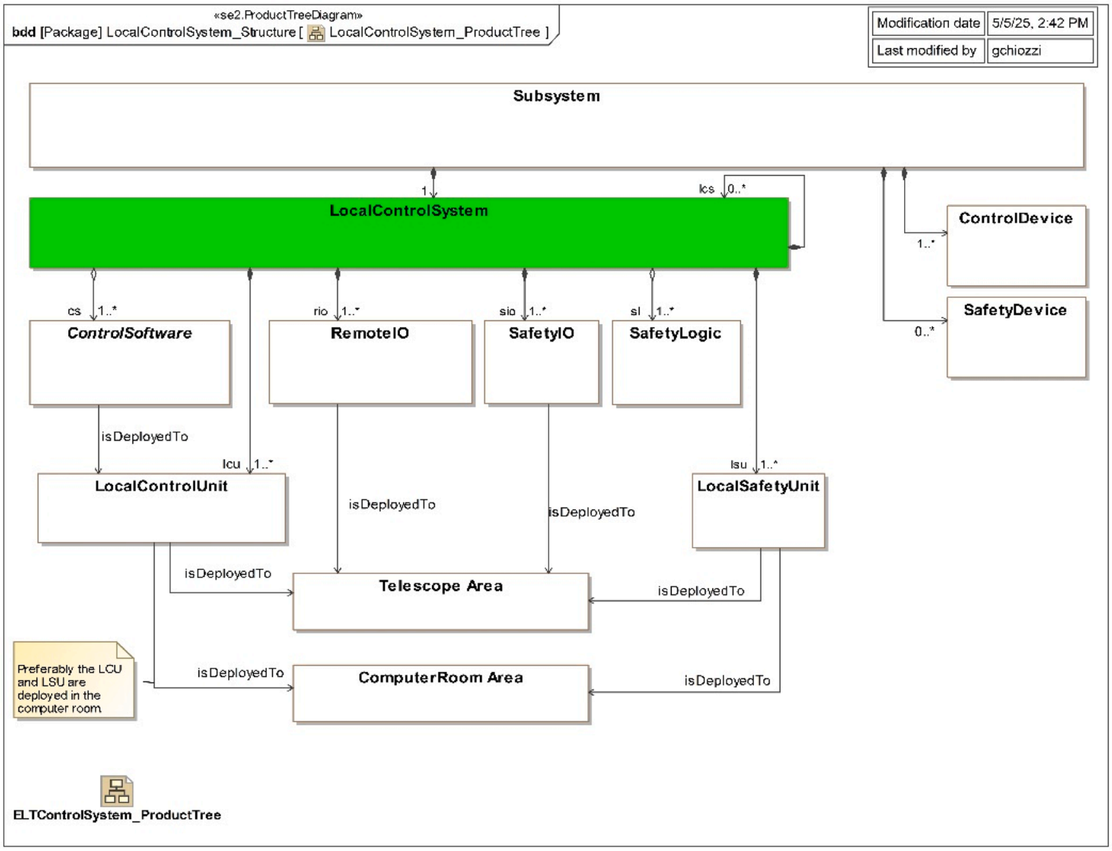

Fig. 4. Local control system architecture.

5. Central control system

5.1. Local supervisors

Each LCS has a partner software component in the CCS called a Local Supervisor (LSV) (Fig. 3-(8)) that is the only interface to the corresponding LCS from other parts of the system: it is the LCS façade.

The LSVs provide several functions: they contain operation- and observation-domain knowledge associated with a subsystem (e.g. tracking a celestial target), they allow for hardware adaptation in case it will be necessary to introduce changes in interfaces or to manage non-compliances in the LCSs emerging during development or AIV. Through design patterns, they implement the control and monitoring of the subsystems. For example, sky tracking algorithms are implemented inside the MS LSV code, which will command the MS LCS to move the axes to a specific (alt, az) position. Another example is M1 LSV that implements the control loop to position the M1 segments.

The LSVs are all subject to a common set of requirements and to a common architecture.

A subsystem is presented to the other components of CCS, and in particular to HLCC, via its Subsystem Functions (defined as the operational activities or purposes for which that Subsystem was designed. For example, tracking a target, imaging a target, correcting wavefront, and so forth).

At the design and implementation level this concept has been strongly modularized, and an application framework has been provided to support the common design across all LSVs.

The Subsystem Functions are implemented within the LSV (either via interaction with the LCS, or within the LSV itself) and the knowledge of the astronomical domain resides here.

As a special case and as can be seen in Fig. 3, the Segment Manipulator and the Local Coherencer LCSs do not have corresponding LSVs, but are integrated into the M1 LSV as individual Subsystem Functions, since this allows for a significative simplification in the implementation.

As already described in (Chiozzi et al., 2022), the LSV common architecture relies on the Estimator-Controller-Adapter pattern (Fig. 5) (Ingham et al., 2004).

Each Subsystem Function can consist of multiple Controllers and Estimators, implemented as individual C++ processes. It exposes an interface implemented by a Primary Controller process and this eventually dispatches requests to other controller processes. The Primary Controller is a single point of access for the function, responsible for controlling the whole Subsystem Function. The public interface of the primary controller can be used by the HLCC to control the associated Subsystem Function.

Estimators publish the estimated state of their associated telescope subsystem function as a set of State Variables.

The term State Variable (SV) refers to an element of the Control System that represents a physical state of the System Under Control (SUC) (Ingham et al., 2004). For example, a limit switch in the SUC will physically be in an opened/closed state and the Control System will use the sensor measurements as evidence to estimate the state of the switch as opened/closed.

Controllers publish Target Variables (TV), that describe the intention of the user as commanded to the Controller: where the State Variable should converge to. For example, a Target Variable contains the target coordinates of the telescope axes, and the associated State Variable is the actual position of the axes.

Target Variables can also be used to transport information on the control system state (e.g. Controllers and, if needed, Estimators).

Whenever possible, there should be a clear association between Target and State Variables.

Target Variables are published by the Controller (and not by the Estimator) because:

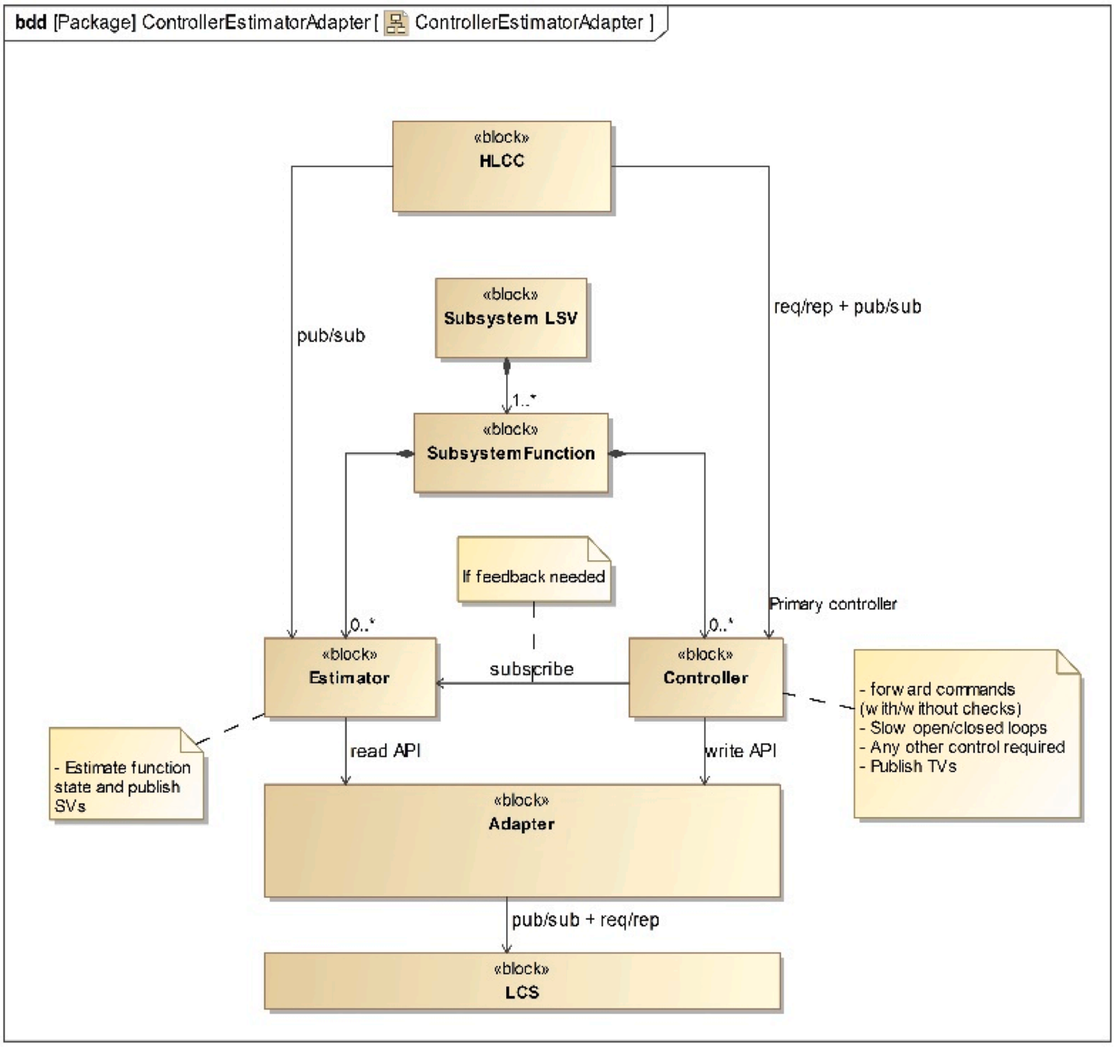

Fig. 5. Estimator-controller-adapter pattern applied to an LSV Subsystem Function.

- The Controller must know the target (which is the set-point for the control algorithm). The Estimator is not required to know the target, on the contrary it should be able to work just with the measurements from the LCS.
- To avoid sending the target to both Controller and Estimator which would have a cost on the HLCC.
- To avoid circular dependencies between Controllers and Estimators, since a Controller may need to subscribe to the Estimator (to receive State Variables).

State and Target Variables are defined as entity data structures and are typically published on DDS and, at low data rates, on the Online Data Base.

All LSV processes, like any other process in CCS, implement a standardized state machine and a set of standard states and commands, to allow building up hierarchically the state of the whole ELT Control System and to allow performing, in a standardized way, common operations like startup and shutdown. The state machine is then extended to the functionality of the specific application (events and transitions).

The state machine is defined using the SCXML notation and is executed in the C++ applications through an interpreter that we have developed (RAD).

Local Supervisors have no safety critical functions, though they may interface with Local Safety Units for observability of the subsystem.

5.2. High level coordination and control

The HLCC offers a single interface for the whole telescope toward operators and the instrument control software (Fig. 3-(3)), except for deterministic interfaces (Fig. 3-(9)) that are provided by the Telescope Real-Time Executor (TREx) (Fig. 3-(4)), as described in section 5.3. The main role of HLCC is to estimate the state of the telescope and coordinate activities across the telescope

HLCC relies on the same basic architectural principles used for the LSVs and uses the same common software infrastructure and libraries, but with some differences motivated by the specific role of HLCC as a telescope high-level supervisor.

The HLCC is architecturally composed of several functional building blocks, as can be seen in Fig. 3. Among these:

- The interface toward the instruments, defined in the CCS-INS ICD, is allocated to the single TelescopeInterface component (Fig. 3-(3)), whose responsibility is to filter, validate and re-dispatch all commands received. This gives flexibility to modify the internal structure without impacting the clients.
- To provide maximum flexibility during AIV and Commissioning, several features of the system and low performance control algorithms are implemented in Python and are managed by the SequencerScripts module (Fig. 3-(10)) or as Jupiter notebooks. Some of these can be executed directly by the operators as independent

applications, while others are embedded in the C++ applications but can be replaced dynamically at run-time using configuration interfaces. In this way, also members of the AIV and commissioning teams can directly modify them without requiring the presence of software experts. This is a lesson learned form the VLT, where such procedures were hardcoded in C++. Some examples are the sequence to execute a Preset of the telescope to a new target, the sequences to startup/shutdown or change operational mode, the procedure to estimate the status of the telescope depending on the status of the different subsystems.

The Sequencer (Munoz, 2022) is a generic ELT tool that allows building Directed Acyclic Graphs (DAG) representing tasks to be carried out (Sequencer scripts). The generated graph defines every task needed (nodes), its order of execution, and its dependencies (edges). Python's asyncio library is used to control and schedule the tasks derived from the DAG. It also allows for pseudo-parallelism between tasks. Despite being asyncio based, the Sequencer is task-agnostic, allowing standard python functions and coroutines to be executed as well. It is composed of various layers: Programmer's API, execution kernel, command line tools and a GUI for interactive execution and manipulation of the graph.

- The Telescope Monitor Status and Configuration component (Fig. 3-(11)) is responsible for monitoring all components of CCS and consolidating this information to build global status indicators. It manages requests for high-level changes in the configuration, propagating information to individual subsystems. It makes high-level status information available to human users as well as to other systems, instruments in particular.
- The PointingKernel (Fig. 3-(12)) component coordinates and interacts with the pointing kernel components in dome, main structure, PFS and laser systems, to which it provides a common library of consistent pointing kernel functions. It interacts with TREx for most of the pointing logic, but also commands the LSVs directly.
- The CentralFDIR (Central Fault Detection Isolation and Recovery) application monitors those aspects of quality and failures that involve more than a single subsystem (Fig. 3-(13)).
- Other dedicated applications exist for star catalogues (like searching for suitable guide stars given a target), monitoring and configuration (like estimation of specific states of the telescope, across several subsystems), alarms, or specific tasks such as M1 segment exchanges (i.e. the segment exchange manager procedure used to replace 2 segments per day for cleaning and re-coating).

Each of the HLCC components in Fig. 3-(14) is implemented as a single C++ process and both estimators and controllers are inside the same process.

The deployment and the lifecycle of all processes in CCS are managed through HashiCorp Nomad and Consul[1]:

- Nomad is a workload orchestrator that can deploy and manage applications across multiple machines, allocating them to the machine that better satisfies the requirements and restarting or relocating them if necessary.
- Consul is a service networking solution that includes service discovery and configuration functionality. In our context, we are using it mainly for service discovery, allowing services to find each other without hardcoding network details.

5.3. Telescope real-time executor

Both HLCC and the Telescope Real-time Executor (TREx) (Fig. 3-(4)) implement telescope-level control tasks. While HLCC addresses supervisory monitoring and slow real-time (up to 20Hz), TREx addresses on-sky loops running at rates up to 1kHz.

TREx receives sensor data from the PFS guide probes and WFS cameras, as well as from the Phasing and Diagnostic Station (Pettazzi et al., 2024), cameras, or post-focal AO modules (the PDS WFRTC or an Instrument AORTC). TREx produces simultaneous commands to the M4, M5, MS and M1 units to achieve system level goals of image stability and image quality, while assuring the M4 Adaptive Mirror always has sufficient stroke.

TREx has several key operation modes:

- Perform field stabilization or guide probe AO (in the order of 50 low order modes) based on PFS Guide Probe (GP) Imager or WFS data. The Phasing and Diagnostic Station Shack-Hartmann WFS may also be used as a guiding sensor.
- Support instrument-driven AO, where tip-tilt control is shared between instrument and PFS sensor data, while the instrument controls the higher order AO modes (low temporal frequencies) (Cascade mode).
- Support instrument-driven AO, where tip-tilt and higher order modes are fully driven by the instrument AO module (Sequential mode).

In all cases TREx performs saturation management of the M4 Adaptive Mirror and M5 Tip-Tilt stage through transfer of accumulated stroke to the Main Structure and M1.

TREx is a multicore datacentre-class Linux server. The software task affinities, interrupts and network stack and hardware ensure low-latency, deterministic reception and transmission of UDP/IP data packets over 100GbE. The UDP/IP data packets follow the ELT "MUDPI" data format (ESO). According to our performance measures we do not need a real time kernel, but proper tuning of the Linux kernel 6.12 is sufficient to satisfy the requirements.

TREx, where possible while meeting latency requirements, will use a configurable software framework and a suite of building blocks to abstract domain logic from underlying real-time computing infrastructure. After trials of several products, GNU Radio[2] has been selected. GNU Radio operates on a flow graph composed of interconnected blocks with streams of items flowing along the connections. Blocks encapsulate the processing of items. They can be hierarchical, i.e. contain more interconnected blocks themselves, which enables nested flow graphs. GNU Radio is used for scheduling using the flowgraph features, but we are not using the built-in blocks. Instead, we use custom out-of-tree blocks, implemented in C++, and operating on a single streaming sample at a time. All network I/O is performed in the TREx framework outside the flowgraph.

5.4. Wavefront commissioning tool

The commissioning interface of the telescope is the Wavefront Commissioning Tool (WFC, Fig. 3-(15) and Fig. 6). The WFC Tool provides high flexibility and script-ability to the commissioning team through the development of Sequencer scripts, Jupyter notebooks or plain Python scripts, such as: M1 segment stacking, pupil registration and photometric sub-aperture calibration with SHAPS (High-order Shack-Hartmann Phasing Sensor). The tool is a Python-based abstraction of all relevant CCS interfaces (PyCCS). Running on a powerful server, it implements telescope simulations, performs computations, data reduction and generates control parameters for the subsystems.

It is the WFC Tool that defines the telescope. This process involves the following general series of steps:

- From the Engineering Archive, load in the WFC Tool the current telescope hardware configuration (M1 segment population, M4 adaptive mirror actuator status, calibration data, etc.).

[1] https://www.hashicorp.com/en

[2] https://www.gnuradio.org/

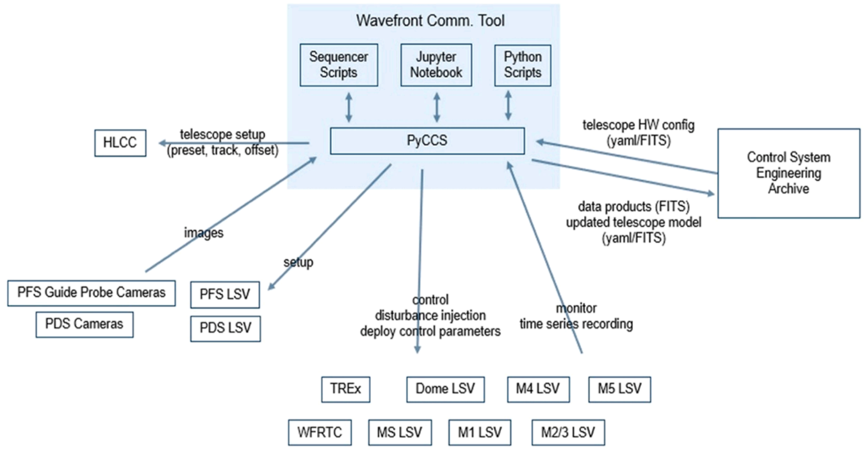

Fig. 6. The wavefront commissioning tool.

- Setup the telescope as required for the commissioning task, e.g. preset and track a natural guide star near the pole, setup Pre Focal Station or the Phasing and Diagnostic Station sensors, etc. This is typically performed through HLCC using interactive command sequences executed under the control of the Sequencer (Fig. 3-(10)).
- Commence the commission task: this is performed via direct interaction with subsystem LSVs and camera systems, commanding offsets to actuators, triggering disturbance injection in controllers, recording loop data and sensor statistics, and triggering exposures on the various telescope cameras. All data is time-stamped at the source, and data correlation is performed via these timestamps. Ensuring the subsystem interfaces can support the functions required by the WFC Tool is an on-going process, taking requirements from various activities: simulation activities, the MELT (Miniscule ELT) test bench in Garching (a small replica of the ELT optical design) and phasing experiments at GTC, a segmented telescope. It is also an on-going activity to work the subsystem interfaces back into the MELT software and simulation models, to ensure the work is compatible with the interfaces.
- The WFC Tool generates control solutions for the telescope: phasing solutions for the M1, M4-to-M1 interaction matrices, calibration files for data reduction, etc. This set of data artifacts is termed the "telescope definition", and is stored, version-controlled, in the Engineering Archive.
- The WFC Tool generates subsystem-specific control and configuration data, from a loaded telescope definition. These are pushed to various subsystems (TREx, M1, M4, WFRTC, etc.). It is the responsibility of the WFC Tool and its versioning system to ensure the control and configuration data is coherent across the full control system. These data contain version information linking it to source.
- The control software will apply the control parameters pushed from the WFC Tool. The version of the telescope definition is written to the Online Data Base, and checked against the configuration loaded by each subsystem. This way, WFC Tool will validate subsystems configuration and create a report with the result..

6. The instrumentation control system

An Instrument Control System (ICS) consists of a combination of hardware and software components providing the functions to control and operate all its own subsystems and enable the acquisition of scientific images for each of the different instrument modes.

The ICS is directly responsible for the control and monitoring of technical and scientific detectors and instrument functions, as well as for the coordination of external systems like the telescope and the post-focal AO module.

The ICS interacts with its users through graphical interfaces designed to automate the observations and to continuously display the status of the instrument and its subsystems.

As illustrated in Fig. 7, the ICS is typically composed of the following subsystems:

- Cabinet Management
- Cryogenic Control
- Local Safety Unit
- Detector Control
- Co-rotation Control
- Built-in AO Control System (AOCS)
- Function Control
- Supervision System
- Maintenance System

ESO provides instrumentation consortia with an Instrument Control System Framework (IFW). This toolkit assists developers in implementing control systems. By offering ready-to-use components, it streamlines software development, abstracts underlying technology, and ensures uniform solutions across all instruments (Kiekebusch, 2024).

The most important ICS FW design principles are:

- Customizability: generic software components can be tailored to the need of the instruments by means of adapting configuration parameters.
- Extensibility: framework components can be extended by developers overriding or specializing the framework solutions to provide instrument specific functionality.

The IFW comprises several components that offer domain-specific frameworks in various areas of control, addressing the diverse needs of instruments. The most important ones are:

- Function Control Framework (FCF): a specific framework for controlling and monitoring instrument hardware functions and sensing systems.
- Camera Control Framework (CCF): a specific framework for controlling both technical and commercial off-the-shelf wavefront cameras.
- Data Display Tool (DDT): a collection of libraries and applications used for the transfer, display and manipulation of data coming from the ELT telescope and instruments.
- Observation Coordination System (OCS): a specific framework for the implementation of supervisory systems and data acquisition.

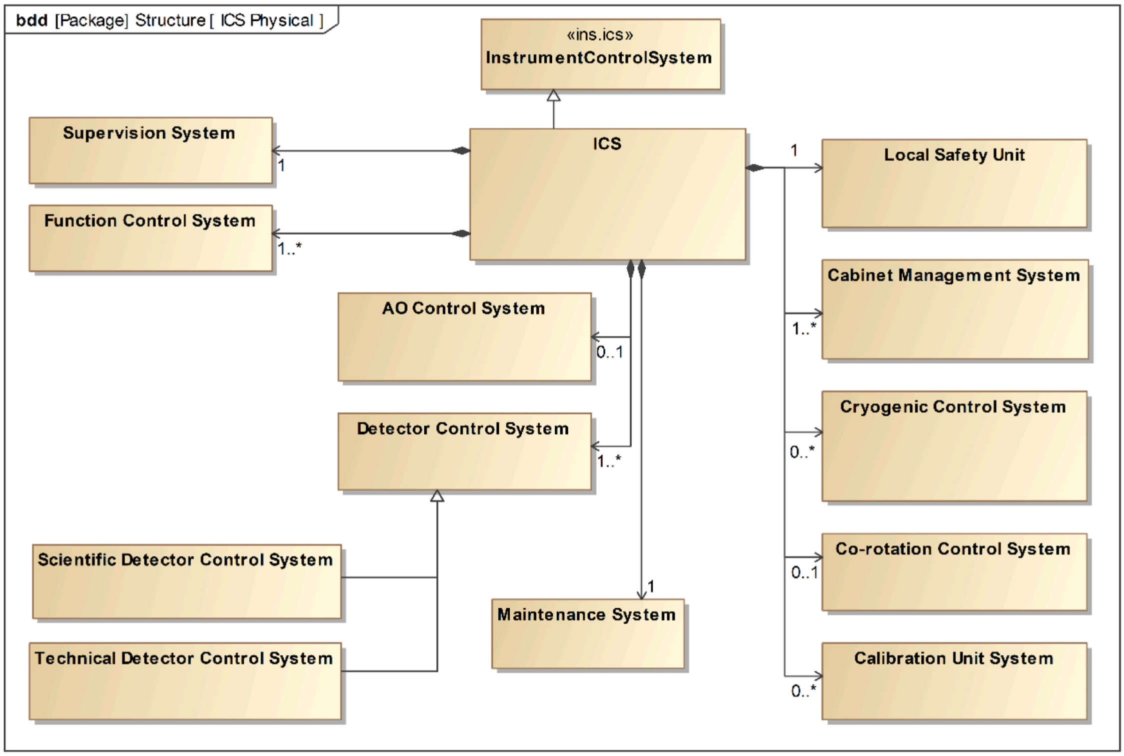

Fig. 7. Instrument control system components.

- Calibration Framework (CalOb): a tool to generate the calibration data needed to process the Science Observations and to monitor the health of the instruments.
- Cryogenic Controller Toolkit (CryoTk): A toolbox for controlling standard cryogenics components like pumps, valves, heaters, coolers, etc., in addition to include the control logic for generic subsystem sequences like evacuation and re-pressurization.

The AOCS is responsible for Instrument's AO Wavefront Correction. The AORTC is the component of the AOCS primarily concerned with the implementation of the main AO loops and it uses the following sensors and actuators in real-time:

- Instrument's AO Wavefront Sensors (WFS)
- Instrument's Focal Plane Sensor (FPS)
- Instrument's AO Corrective Optics
- ELT AO Mirrors M4 and M5

The AO RTC is composed of Hard Real-Time Core (HRTC) and Soft Real-Time Cluster (SRTC).

The main functionality of the HRTC is:

- Calculating and applying AO corrections at rates in the 100Hz to several KHz range.

The main functionality of the SRTC is instead:

- Receiving telemetry data from HRTC.
- Optimization of the HRTC operation.
- (re)configuration of the HRTC.
- Commanding and Monitoring the HRTC.
- Interacting with the other components of the instrument (ICS).

The ELT RTC Toolkit (Jeram) is a suite of software tools, libraries, troubleshooting utilities, simulators and example implementations for the development of the SRTC of an ELT-scale AORTC. The toolkit is being developed by ESO to help telescope and instrument RTC developers implementing their systems.

7. Operation workflow

The diagram in Fig. 8 takes key components from the ELT Control System (CS) and places them in an interface block diagram where the interfaces relevant to the telescope during observation are identified.

The Main Structure (MS), Dome, PFS and Laser Guide Star Facility (LGSF) LSVs track sky coordinates and thus run pointing kernels publishing actuator position references at 20Hz. The MS (Alt/Az axes), and PFS (guide probes) may receive deterministic input into the pointing kernel in the form of adjustments to the target position on the focal plane (from the instrument). TREx, as part of guiding and stroke management, computes and publishes tip-tilt commands to M4, M5 and MS, via a collaborative control scheme (Sedghi, 2018) in which each actuator receives the same goal but tries to achieve it over its own characteristic bandwidth.

Such control-sharing strategies aim at maintaining the residual wavefront error within requirements.

Low order modes accumulating on M4 (and in large parts originating in quasistatic deformations of the telescope) are transferred to M1. The outcome of this transfer is the likely storage of focus and coma on M1 when it belongs on M2. For the purposes of compensating this error, ROUS (Recurrent Optimization of M4 Unit's Stroke) maneuvers are employed when it is deemed necessary (at time intervals $\geq$ 5 minutes) and after having notified the observing instrument that has a chance to prevent the maneuver, since it can affect the observation.

The M4 Adaptive Mirror and M5 Tip-Tilt stage are driven by TREx at rates up to 1kHz. To minimize latency, the commands from TREx are published directly to the M4 and M5 Local Control Systems (LCSs), bypassing the respective LSVs.

System-level conditions of the telescope are estimated from the set of subsystem Target and State Variables. These conditions are summarized by the Telescope Monitor Status and Configuration, for example: "presetting", "guiding", "ready for INS hand-over", "observing". Such

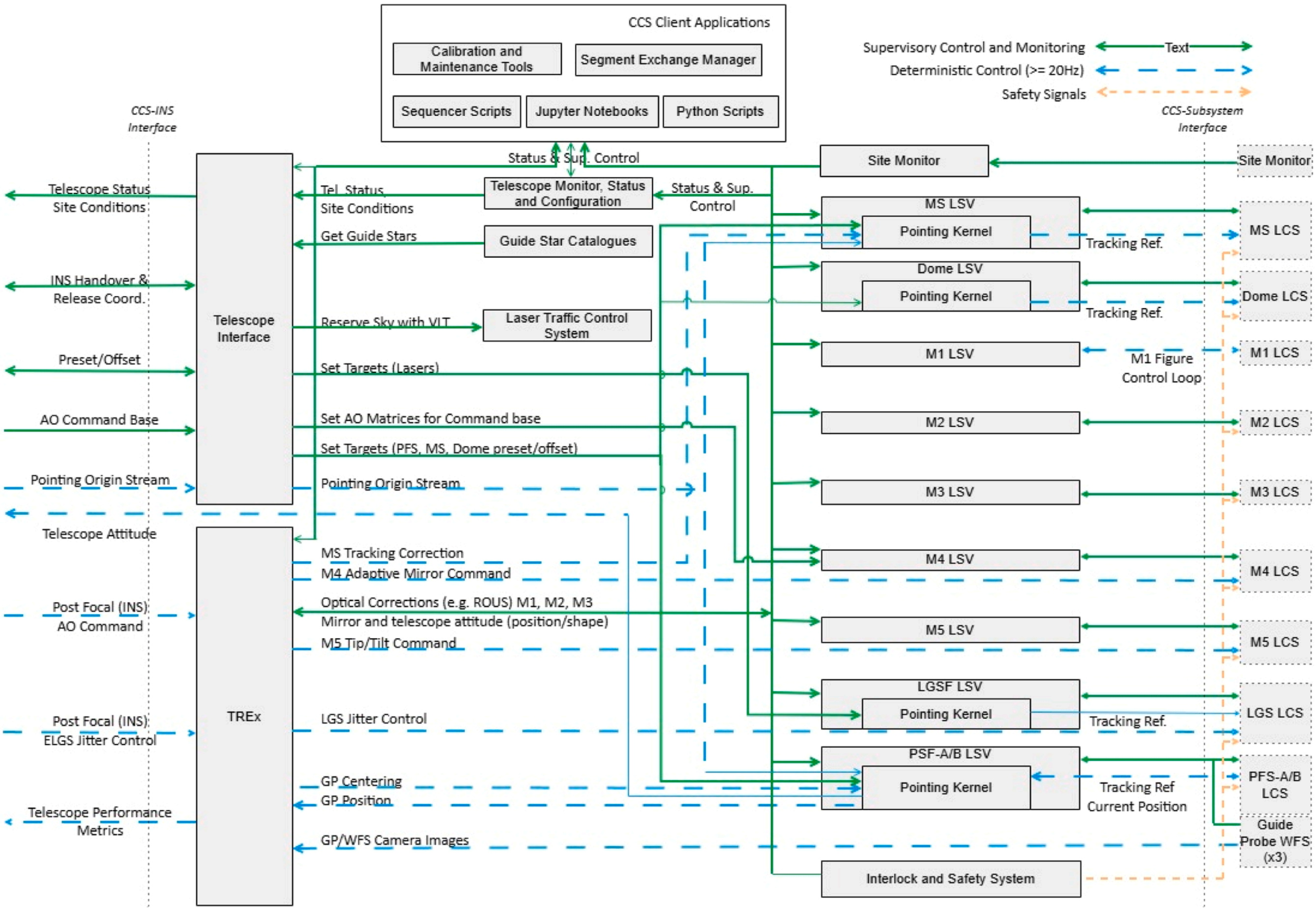

Fig. 8. CCS interfaces.

conditions are not mutually exclusive, and they do not inhibit the function of the telescope, rather they are used to inform the operators and coordinate observations with the instruments.

ESO technical staff (i.e operators, AIV and commissioning scientists and engineers, maintenance engineers or the software development team) can interact with the telescope using the operator consoles and engineering GUIs or through Python scrips run through the Sequencer or Jupyter-notebooks.

At the start of the night, the telescope subsystems are initialized and transitioned into night operations mode by the operators using the Sequencer.

With the dome's slit doors open and the necessary subsystem functions available and enabled, the telescope is ready to observe.

The primary users of the telescope are the instruments and, through them, the astronomers preparing the observations to be run.

An instrument commences using the telescope by sending the observation specific parameters to the CCS: target, natural guide stars, observing wavelength, AO command base, etc, after which it commands the telescope to Preset. The guide probe cameras are set up according to the expected natural guide star characteristics. TREx is configured as to which guide probe will be used for guiding, the AO command base, and other observation parameters. Likewise, the M4 Adaptive Mirror is loaded with the AO command base.

The telescope interface, upon reception of preset, will forward the preset coordinates to Dome and Main Structure pointing kernels, running in the respective LSVs, which commence publishing position and trajectory demands at 20Hz to the telescope and dome axes (driven by the corresponding LCSs). The PFS LSV will receive coordinates of up to three natural guide stars (queried from the catalogue, if not provided by the observation) and will likewise commence publishing position and trajectory data to the guide probe arms (driven by the PFS LCS), and properties of the sky focal plane image (direction of celestial north, direction of zenith, plate scale, coordinates at reference location...) at the INS-CCS interface for consumption by instrument tracking functions such as rotators, pupil steering mirrors and pointing origin stream. At the time of preset, the instrument will commence publishing its pointing origin stream (i.e. the coordinates that define the projection of the instrument's derotator center onto the PFS straight through focal plane), which is received by the Main Structure and PFS pointing kernels.

When the guiding-designated guide probe (typically that on the brightest star) images the natural guide star, the exposure (made available to the operator via TREx) may be adjusted through usual camera controls such as integration time and windowing, to improve the image quality. In addition to image display, TREx produces image quality metrics, tip, tilt, and rotation error estimates based on the available natural guide stars and publishing these to the CCS (and INS interface). After the natural guide star selected by the operator is centered on the guide probe reference pixel, TREx begins computing and publishing tip-tilt commands to M4, M5 and MS and guiding commences, with the loop closed on the brightest natural guide star.

Once guiding, TREx will use sampled images and telescope metrology (published as LSV State Variables) to calculate periodic corrections to the telescope optics. The primary goal of these corrections is to offload lower frequency, but higher amplitude required strokes from the M4 Adaptive Mirror, while improving image quality to a level suitable for instrument AO to begin. The instrument is receiving light at all stages and therefore it may commence its setup and acquisition process, but the telescope is not committed to deliver and maintain a defined image quality or to accept requests from the instrument in this optimization phase. A drawback of this strategy is that the instrument is responsible for protecting its own equipment: as an example, IR detectors must be protected from excessive light exposure producing “image persistence” that would affect the image quality of the following observation. After one or more iterations, the CCS will indicate “ready for handover” to the instrument, at which time the instrument may conclude its acquisition and is ready for AO.

The instrument performs its acquisition procedures concluding in a deterministic signal exchange between the instrument AO RTC and TREx, whereupon TREx switches from guiding (based on errors derived from the PFS guide probes), to processing of the instrument-published AO commands. The handover lasts a few milliseconds at which time no corrections are sent to M4 adaptive mirror, M5 tip-tilt and MS axes (MS is still tracking). After successful handover the instrument is performing guiding and higher-order AO corrections, and TREx is responsible for mirror stroke optimization, primarily through its collaborative control scheme. The PFS guide probes continue tracking the natural guide stars; the guiding corrections derived from the PFS natural guide star may or may not be incorporated into the collaborative control scheme by TREx depending on the mode selected by the instrument: in “Cascade” mode the guiding corrections from the guide probes are incorporated with guiding corrections from the instrument; in “Sequential” mode, the PFS-derived guiding errors are ignored with tip-tilt and higher order AO modes fully driven by the instrument.

At any time after handover, whether requested by the instrument, the telescope operator, or as the result of an internal condition (e.g. optical unit saturation), the telescope may ignore (not apply) instrument AO corrections and switch back to the PFS-derived guiding corrections. These conditions are communicated to the instrument both over the deterministic network (to the instrument AO RTC) and the control network (to the instrument supervisory software), and instrument acquisition may be repeated.

The transition through acquisition to CCS-driven wavefront control onto INS-driven wavefront control and back is shown in the activity diagram in Fig. 9.

8. Conclusions

At this stage of the project, we have consolidated and tested performance and reliability of the software platform, implemented foundational design patterns and core functionality of LSVs and CCS control coordination components.

Moreover, several subsystems have already undergone preliminary acceptance, have been delivered, or are currently undergoing test campaigns.

In parallel, we have begun the integration and joint testing of all control system elements on real hardware at ESO HQ. This hardware has been supplied either by the industrial contractors developing the LCSs or we have implemented dedicated test benches in-house, like the Miniscule ELT. These early integration activities are critical to validating interfaces, refining software interactions, and mitigating risks in upcoming deployment phases at the telescope site.

At the same time, consortia are developing the control system of the respective instruments and we can test the integration with the telescope in Garching.

More details on the status of development of each LSV and of the other components are available in (Chiozzi, 2024).

CRediT authorship contribution statement

Gianluca Chiozzi: Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Luigi Andolfato:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Javier Argomedo:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Nicolas Beneš:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Carlos Diaz Cano:**

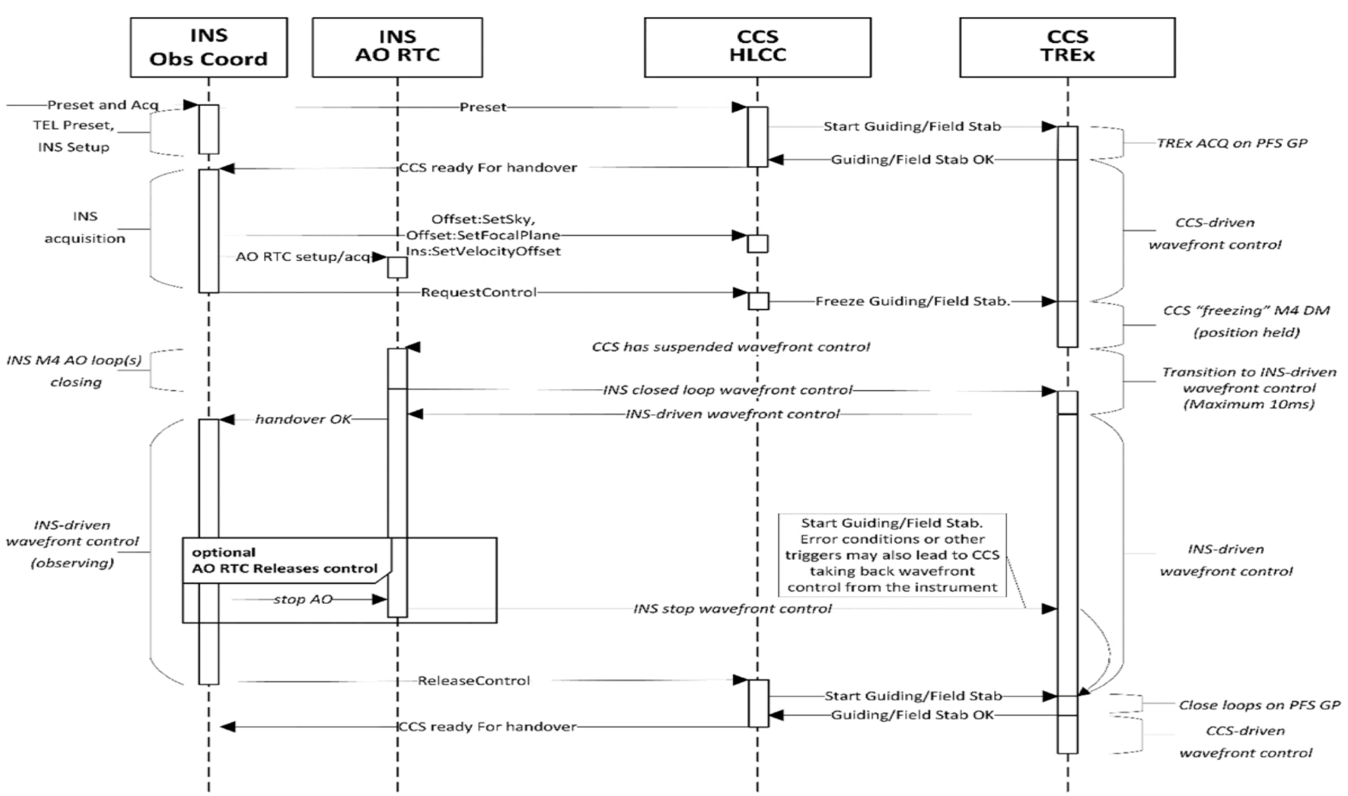

Fig. 9. Transition through acquisition to CCS-driven wavefront control onto INS-driven wavefront control.

Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Robert Frahm:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Josef Hofer:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Bogdan Jeram:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Mario Kiekebusch:** Writing – original draft, Writing – review & editing. **Nick Kornweibel:** Writing – review & editing, Writing – original draft, Supervision, Software, Investigation, Conceptualization. **Ulrich Lampater:** Writing – review & editing, Writing – original draft, Investigation, Conceptualization. **Federico Pellegrin:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Andres Ramirez:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Marcus Schilling:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Babak Sedghi:** Writing – review & editing, Writing – original draft, Investigation, Conceptualization. **Heiko Sommer:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization. **Arturo Hoffstadt Urrutia:** Writing – review & editing, Writing – original draft, Software, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank all members of the extended ELT team—both ESO staff and contractors—whose contributions have been essential to the work presented in this paper. Their continued efforts in building the ELT are instrumental in realizing its potential as a breakthrough scientific facility.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ascom.2026.101171.

References

Biasi, R., et al., 2016. E-ELT M4 adaptive unit final design and construction: a progress report. In: Proc. SPIE 9909, Paper 9909-7Y.
Chiozzi, G., et al., 2018. Designing and managing software interfaces for the ELT. In: Proc. SPIE 10707, paper 10707-78.
Chiozzi, G., et al., 2022. The ELT high-level coordination and control. In: Proc. SPIE 12189-48.
Chiozzi, G., et al., 2024. Status of the ELT control software development". In: Proc SPIE 13101, Paper 13101-4.
ELT ICS Rapid Application Development (RAD), http://www.eso.org/~eltmgr/ICS/documents-latest/RAD/sphinx_doc/html/.
ESO-302020 - MUDPI Standard, internal ESO ELT project document .
Hoffstadt, A., et al., 2024. Taurus integration to ELT control software. In: Proc SPIE 13101-3. https://doi.org/10.1117/12.3019717.
Ingham, M., et al., 2004. Engineering complex embedded systems with state analysis and the mission data system. In: Proceedings of First AIAA Intelligent Systems Technical Conference. https://trs.jpl.nasa.gov/handle/2014/38225.
Jeram, B. et al., "RTC toolkit - ESO's solution to address challenges of implementing new-generation RTCs", Proc. SPIE 14155-62.
Kiekebusch, M.J., 2024. Advancements in astronomical instrumentation: a new control software framework for ELT and VLT instruments at ESO. In: Proc. SPIE 13101-17.
Munoz, I., 2022. The ELT sequencer. In: Proc SPIE 12189, Paper 13101-4. https://doi.org/10.1117/12.2629596.
Pellegrin, F., Rosenquist, C., 2017. The ELT linux development environment. In: Proc. 16th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain. https://doi.org/10.18429/JACoWICALEPCS2017-THBPL05. Oct.
Pettazzi, L., et al., 2024. Status of the ELT phasing and diagnostic station project. In: Proc. SPIE 13099, paper 130990J.
Reinacher, A., et al., 2024. Advancements of the ELT M1 fig. control loop: edge sensor fault detection and management of actuator non-linearities in a large MIMO system. In: Proc. SPIE 13094.
Sedghi, B., 2018. Tip/tilt control strategies at ELT. Wavefront sensing and control in the VLT/ELT era, 3rd edition. invited talk, Paris. https://indico.obspm.fr/event/56/contributions/145/.
Tamai, R., et al., 2026. ESO's ELT takes shape: first deliveries and start of AIV. In: Proc. SPIE 14147-46.