Designing and prototyping the control system for the Cherenkov Telescope Array

I Oya[1] M Füßling[1] P O Antonino[2] M Araya[3] J Borkowski[4] A Bulgarelli[5] J Castroviejo[6] P Colome[6] V Conforti[5] A Garcia[6] L Hagge[7] E Lyard[8] A Morgenstern[2] M Mayer[9] D Melkumyan[1] T Murach[1] L Pizarro[3] I Sadeh[1] T Schmidt[1] U Schwanke[9] G Spengler[9] J Schwarz[10] G Tosti[11] R Walter[8] P Wegner[1] for the CTA Consortium[12]

[1]DESY, Zeuthen, Germany, [2]Fraunhofer IESE, Kaiserslauten, Germany , [3]UTFSM, Valparaíso, Chile, [4]CAMK, Torun, Poland, [5]I.A.S.F. di Bologna, Italy, [6]IEEC-CSIC, Spain, [7]DESY, Hamburg, Germany, [8]University of Geneva - Departement d'Astronomie, Switzerland, [9]Humboldt-Universität zu Berlin, Germany, [10]INAF – Osservatorio Astronomico di Brera, Italy, [11]University of Perugia, Italy [12]See www.cta-observatory.org

ABSTRACT

The Cherenkov Telescope Array (CTA) is the next-generation atmospheric Cherenkov gamma-ray observatory. The Observation Execution System (OES) team within the CTA project is designing and prototyping the software to execute the observations and to handle the acquisition of scientific data at GB/s rates. In this contribution we show the OES system as it is being designed using the UML and SysML formalisms. In addition, we present the status of the associated prototyping activities.

Supported by:

THE CTA ARRAY

CTA will consist of two facilities, one in the southern (Cerro Armazones Chile) and the other in the northern hemisphere (La Palma, Spain). The two sites will contain dozens of telescopes of different sizes, constituting one of the largest astronomical installations under development.

CTA OPERATIONS

CTA will implement simultaneous automatic operation of multiple sub-arrays. It will be capable of quick re-scheduling of observations (within a few seconds), in order to allow observations of elusive transient events. The operation, control, and monitoring of the distributed multi-telescope CTA arrays is inherently complex. As such, they pose new challenges in scientific instrumentation control systems and in particular in the context of ground-based gamma-ray astronomy.

MODEL DRIVEN APPROACH

The OES architecture (see Fig. 1) is designed using the Software Platform Embedded System (SPES) methodology [1]. The application of the architecture model includes, among others:

- Architecture drivers and design using the viewpoints defined in SPES (Fig. 2)
- Specification of architecture design, including interfaces and behavior
- Relationships (*traces*) from drivers to model elements, ensuring consistency
- Vision sharing and scoping (incl. interface definition), two important ingredients for collaborative software engineering

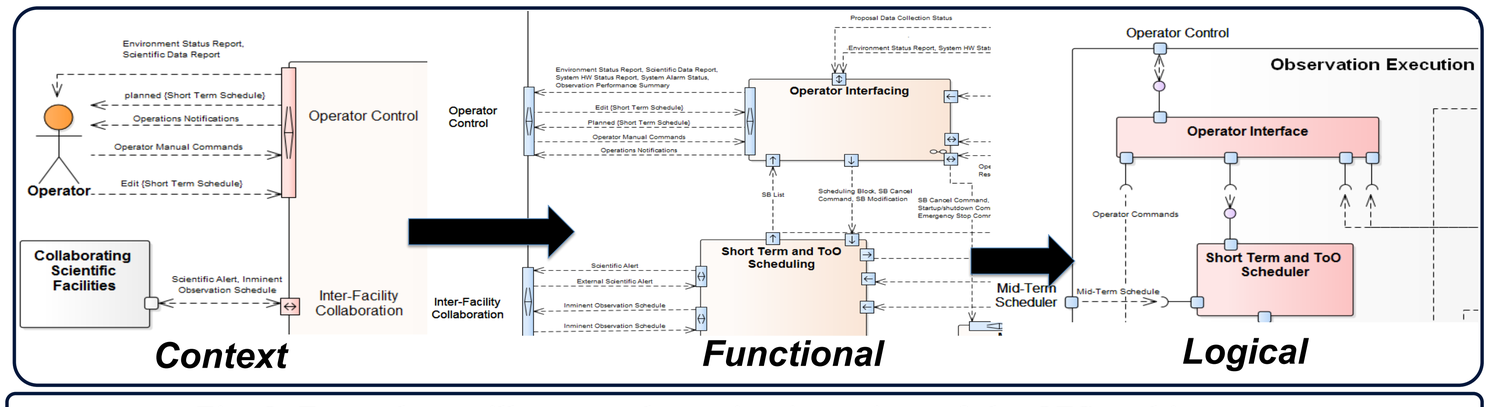

Fig. 2: Examples to illustrate the main viewpoints of the OES architecture.

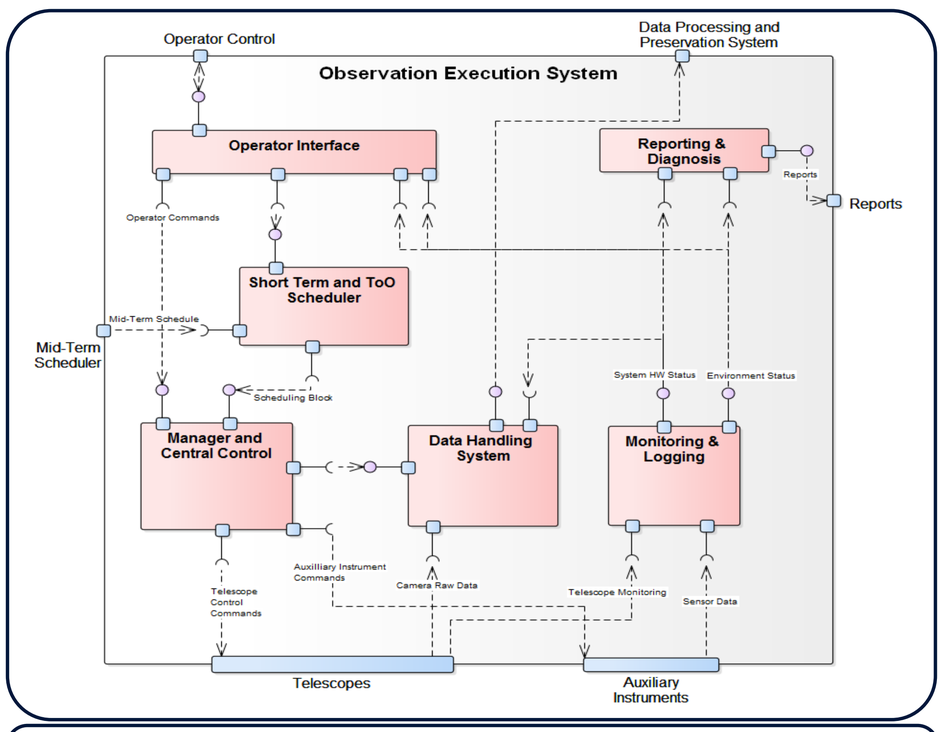

Fig. 1: Logical view of OES, representing the main components of the system. Only the most relevant components, data and interfaces are shown.

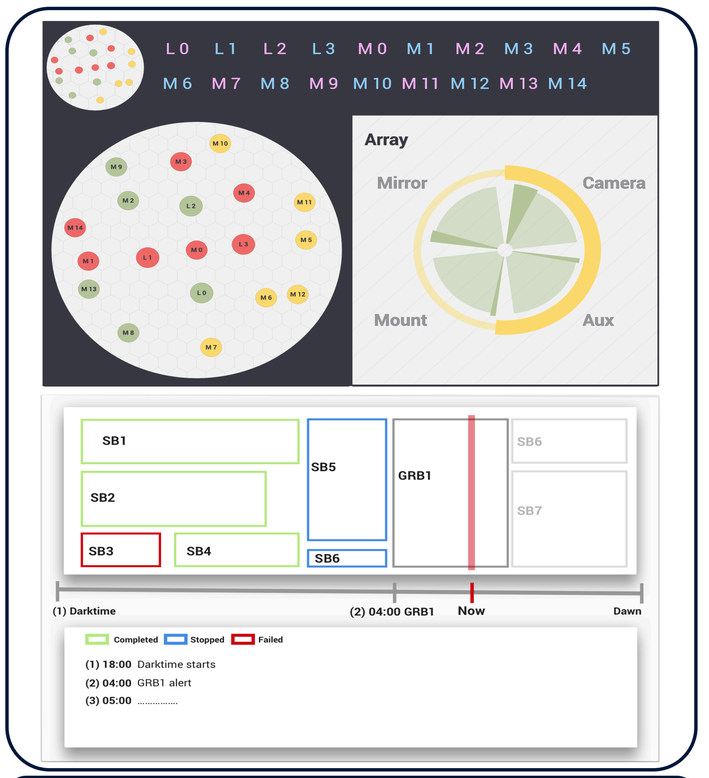

Fig. 3: Two panels from the operator interface prototype, *array zoomer* (upper panel), and *nightly overview* (lower panel).

PROTOYPES

The OES is a distributed software application using the Alma Common Software (ACS) CORBA-based framework. We have produced advanced prototypes for most OES sub-systems:

- Data handling system, containing central trigger, data acquisition, and local storage, in C/C++ [2].
- Manager and Central Control (in Java/ Python) containing the operation scripts library and an implementation of a supervision tree architecture.
- Operator interface [3], in JavaScript and Python (Pyramid), see Fig. 3.
- Short term scheduler, in C++, using advanced artificial intelligence algorithms (metaheuristic optimization) [4].
- Monitoring system, prototyping solutions based on redis, MongoDB and Cassandra data base technologies.

ACKNOWLEDGEMENTS

We gratefully acknowledge financial support from the agencies and organizations listed here: www.cta-observatory.org/consortium_acknowledgments

REFERENCES

[1] Pohl, K. et al. Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology, Springer Publishing, Berlin Heidelberg (2012)
[2] Lyard, E., Walter, R., End-to-end data acquisition pipeline for CTA, ICRC 2017
[3] I. Sadeh, et al. Prototyping the graphical user interface for the operator of the Cherenkov Telescope Array talk Proc. SPIE 9913, 99130X (2016)
[4] Colomé, J., et al., Artificial intelligence for the CTA Observatory scheduler, Proc. SPIE 5496, 205-218 (2012)