The software system for the Control and Data Acquisition for the Cherenkov Telescope Array

P Wegner[1], M Füßling[1], I Oya[1], L Hagge[2] U Schwanke[3], J Schwarz[4], G Tosti[5], V Conforti[6], E Lyard[7], R Walter[7] for the CTA Consortium;

P Oliveira Antonino[8], A Morgenstern[8]

[1] DESY Zeuthen Germany
[2] DESY Hamburg Germany
[3] Humboldt Universität zu Berlin, Germany
[4] INAF - Osservatorio Astronomico di Brera, Italy
[5] University of Perugia, Italy
[6] INAF - IASF Bologna, Italy
[7] ISDC, Univ. Geneva
[8] Fraunhofer IESE, Kaiserslautern, Germany

E-mail: Peter.Wegner@desy.de

Abstract. The Cherenkov Telescope Array (CTA), as the next generation ground-based very high-energy gamma-ray observatory, is defining new areas beyond those related to physics. It is also creating new demands on the control and data acquisition system. CTA will consist of two installations, one in each hemisphere, containing tens of telescopes of different sizes. The ACTL (array control and data acquisition) system will consist of the hardware and software that is necessary to control and monitor the CTA array, as well as to time-stamp, read-out, filter and store the scientific data at aggregated rates of a few GB/s. The ACTL system must implement a flexible software architecture to permit the simultaneous automatic operation of multiple sub-arrays of telescopes with a minimum personnel effort on site. In addition ACTL must be able to modify the observation schedule on timescales of a few tens of seconds, to account for changing environmental conditions or to prioritize incoming scientific alerts from time-critical transient phenomena such as gamma-ray bursts. This contribution summarizes the status of the development of the software architecture and the main design choices and plans.

1. Introduction

This paper gives a short overview on the Cherenkov Telescope Array (CTA) project [1] and then describes the software system for the CTA Array Control and Data Acquisition work package (ACTL).

The CTA project is an initiative to build the next generation ground-based very high-energy gamma-ray instrument. CTA opens a new era in gamma-ray astronomy. It will serve as an open observatory to a wide astrophysics community and will provide deep insight into the non-thermal high-energy universe. The present generation of imaging atmospheric Cherenkov telescopes (H.E.S.S., MAGIC and VERITAS) has in recent years opened the realm of ground-based gamma-ray astronomy in the

energy range above a few tens of GeV. The CTA design foresees a factor of 5-10 improvement in sensitivity compared to the current very high-energy gamma-ray domain. It needs to be optimized for a wide energy band, which leads to the design of telescopes of three different sizes.

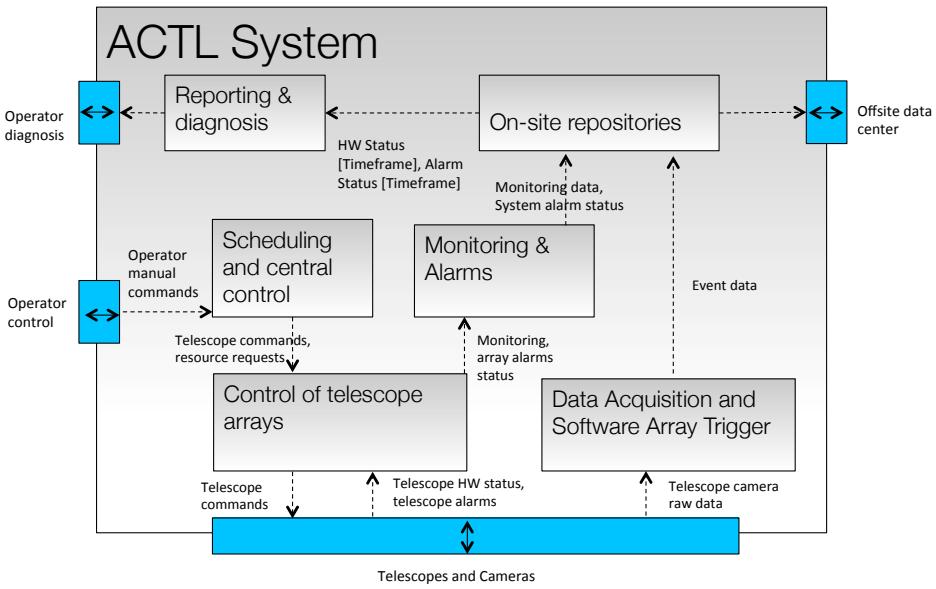

Figure 1. ACTL building blocks

CTA will consist of two arrays for complete sky coverage: a southern array consisting of about 100 telescopes and a northern array with about 20 telescopes. Detailed contract negotiations have been started with the southern site candidate, which is located at less than 10 km southeast of ESO's existing Paranal Observatory in the Atacama Desert, and the northern site candidate, which is located on the existing site of the Instituto Astrofisica de Canarias Observatorio del Roque de los Muchachos on the island of La Palma. Sites in Namibia (south) and Mexico (north) will be kept as viable alternatives.
The CTA Consortium consists of over 1200 members from more than 200 institutes in 32 countries.

2. Array Control and Data Acquisition – ACTL

The system to control, monitor and readout the CTA array (Array Control and Data Acquisition system, ACTL) will provide means for (see Fig. 1 for a schematic view):

- Data Acquisition (DAQ)
- Central control and execution of observations
- Control and monitoring of all Cherenkov telescopes and auxiliary devices
- Automatic scheduling of observations
- Central array triggering and nanosecond time synchronization
- On-site repositories for data storage and an environment to execute a real-time analysis

The much larger size and complexity of CTA compared to current installations lead to numerous challenges for ACTL:

- Boundary conditions involve much larger latencies, more difficult synchronization, higher data rates, larger data volume and a raw data bandwidth comparable to that of the LHC ATLAS (before the recent upgrade)
- CTA supports more telescopes of different types than present installations, encompassing heterogeneous control and readout architectures

- CTA is intended to act as an open observatory, which requires the provision of high-standard services and very high reliability beyond the levels of current gamma-ray experiments

An advanced and well-conceived software design based on adequate software architecture together with the use of software frameworks and standards are essential ingredients for reaching the required high level of reliability and availability of ACTL products. For all underlying hardware for processing, data storage, networking and triggering, commodity systems are being used.

3. ACTL software model

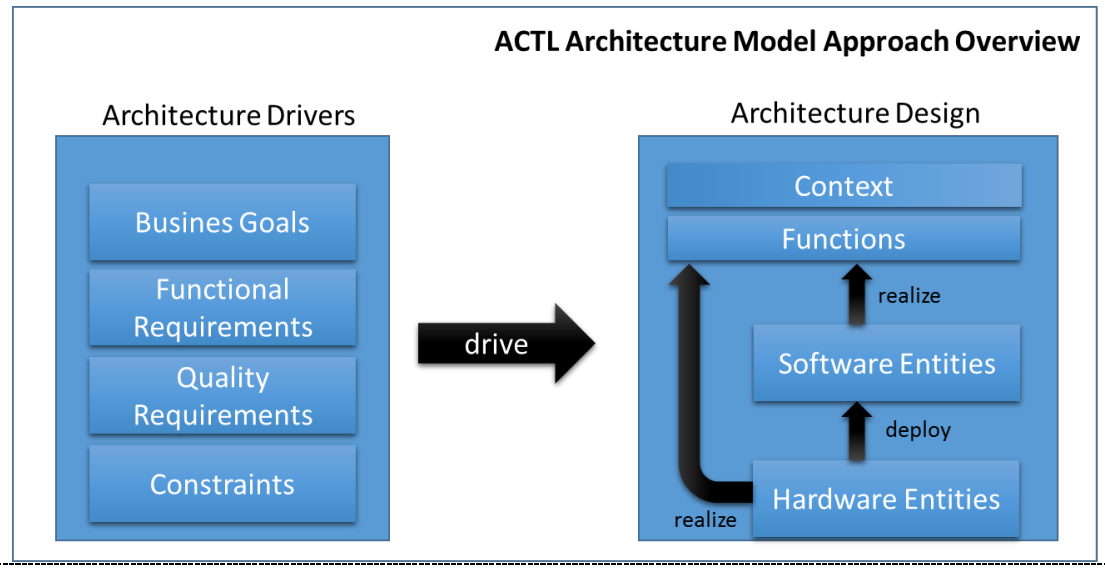

Figure 2. Schematic view of the ACTL system model viewpoints and the associated design steps.

Software architecture can be defined as the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.

A preliminary version of the ACTL architecture model has been introduced and established as a common means of communication within ACTL. The model is being used for establishing and driving the system development, and as foundation for project management. The architecture model provides intuitive schematic visualization of the ACTL system, and allows vision sharing within the ACTL collaboration, as well as with the ACTL stakeholders in the CTA project team and beyond.

The model represents the entire ACTL system. It comprises the drivers for the ACTL architecture (e.g. requirements, specification and use cases) together with a description of the ACTL functionality, and of the software and hardware components that have to be provided (see Fig. 2). The model follows the SPES2020 approach for describing embedded (real-time) systems [2], using a tailored [3] SysML notation (Systems Engineering Modeling Language) [4] and the modeling software "Enterprise Architect" [5].

Figure 3 illustrates one of the diagrams contained in the views of the ACTL architecture model. In the context view, the ACTL system is depicted as a black box (big box in the center), together with the

external elements, which can be either human stakeholders, depicted as stick figures in the diagram, or external systems, which are depicted as the smaller boxes outside the main box. The information exchange between the ACTL system and the external entities are indicated by the Ports (blue boxes) at the boundaries of the ACTL system, along with the directed connections (arrows) and the data elements transported by each connection.

The idea of using a model-based approach aimed at supporting two goals. The first is to define a systematic architecture specification for the entire ACTL project to support:

- Identification of all architecture drivers, such as requirements, use cases and stakeholders, and tracing them to functionalities, and then to software and hardware components.
- Providing a framework for all ACTL product owners to further develop the detailed design of their products
- Providing guidelines to be followed by all ACTL software developers in the development and implementation of their software components and to document each design choice and its implications

On the other hand the architecture model helps us to reach project-related goals, which include:

- Develop an implementation plan and enable progress control
- Identify unplanned work and assign responsibilities
- Evaluate ACTL project priorities, main risks and their mitigation
- Provide means for product acceptance (verification and validation)
- Allow deriving cost and effort estimates for realizing the ACTL system

4. ACTL frameworks

The high-level ACTL software is developed on top of the ALMA Common Software (ACS [6]), a software framework for the implementation of distributed data acquisition and control systems. ACS has been developed by ESO and has been successfully applied in projects of similar scale (e.g. ALMA [7]). ACS distributions (provided by ALMA computing) are executed on a variant of the Linux operating system (RedHat Enterprise Linux re-distribution), which will therefore be used on all full-scale ACTL computers.

ACS is based on a container-component model and supports the programming languages C++, Java and Python (the latter in particular for scripting). The high-level ACTL software will be executed predominantly on the central computer cluster and will access most hardware devices via OPC Unified Architecture (OPC UA, see [8]). OPC UA is an industrial standard, so OPC UA servers can either be provided by device vendors or be developed by the device teams. In most cases, the functionality of the OPC UA servers (variables, methods etc.) defines the interface between ACTL and a hardware device (e.g. a weather station or CCD camera) that must be controlled and read-out.

5. Conclusions

The design of the ACTL software accounts for the need to develop, maintain, and operate a complex but very reliable software system with an efficient use of limited resources. To allow achieving its goals, the ACTL project has adopted a model-driven development dynamic based on the SPES2020 methodology using the SysML notation.

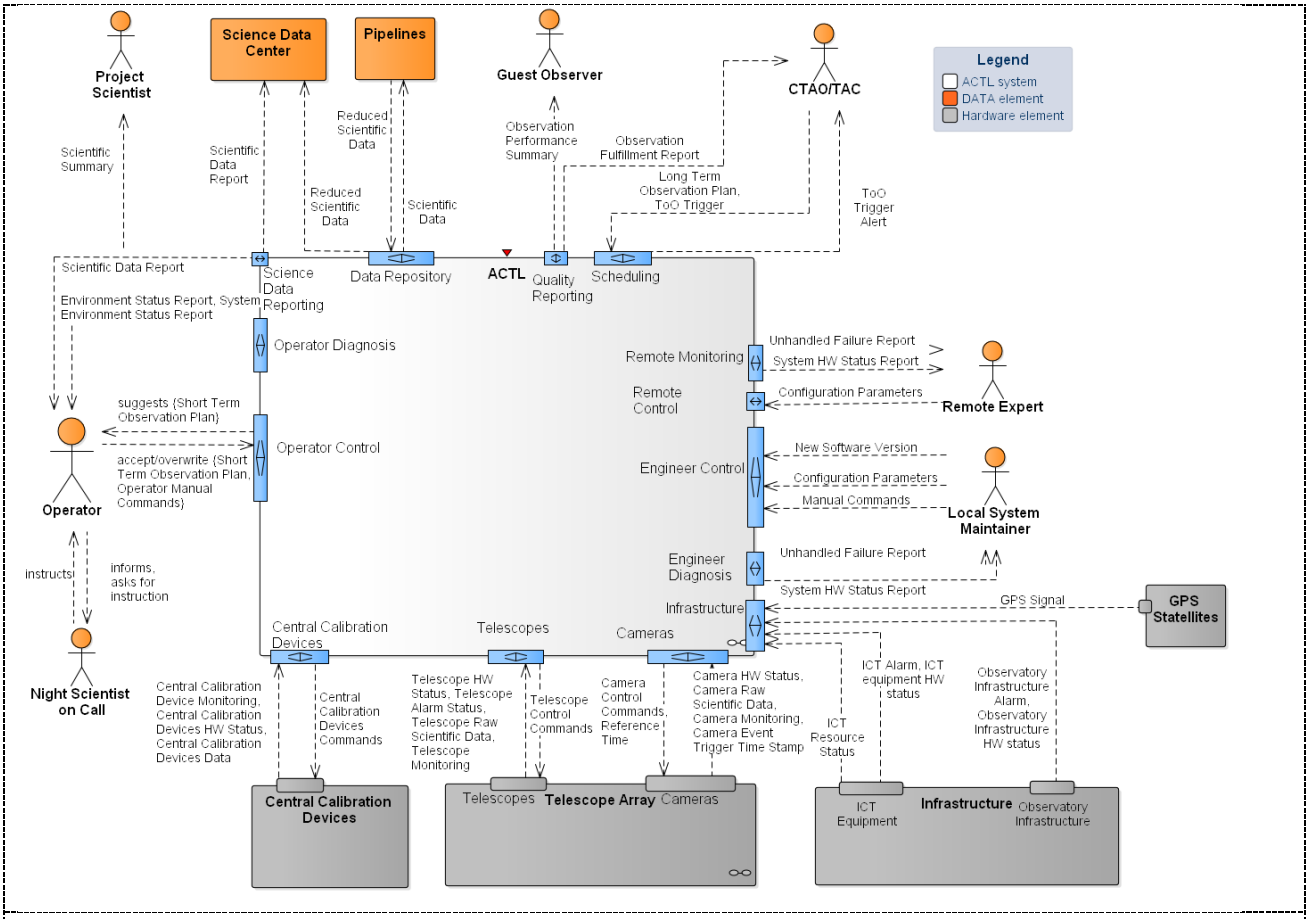

Figure 3. Example element of the ACTL architecture model for CTA: the context diagram. See text for details.

Acknowledgments

We gratefully acknowledge support from the agencies and organizations listed under Funding Agencies at this website: http://www.cta-observatory.org/

References

[1] B S Acharya et al., Introducing the CTA concept, APh 2013 (43) 3.

[2] Klaus Pohl, Harald Honninger, Reinhold Achatz, and Manfred Broy. Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology. Springer, Berlin Heidelberg, 2012

[3] SysML – "OMG Systems Modeling Language 1.4", OMG 2015

[4] Thomas Kuhn, Pablo Oliveira Antonino. Model-Driven Development of Embedded Systems. Embedded Software Engineering Congress 2014. Sindelfingen, Germany, December 2014.

[5] Enterprise Architect – http://www.sparxsystems.com

[6] Chiozzi G, Jeram B, Sommer H *et al*. (2004). *The ALMA common software: a developer-friendly CORBA-based framework*. In H. Lewis & G. Raffi (editors), *Advanced Software, Control, and Communication Systems for Astronomy*, volume 5496 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 205–218

[7] Wootten A & Thompson A R (2009). *The Atacama Large Millimeter/Submillimeter Array*. IEEE Proceedings, **97**, 1463

[8] OPC UA foundation: http://www.opcfoundation.org/