Prototyping the Resource Manager and Central Control System for the Cherenkov Telescope Array.

D Melkumyan[1], I Oya[2], M Fuessling[2], I Sadeh[1], S Sah[3], M Sekoranja[3], T Schmidt[1], U Schwanke[4], J Schwarz[5], P Wegner[1], for the CTA Observatory[6]

INTRODUCTION

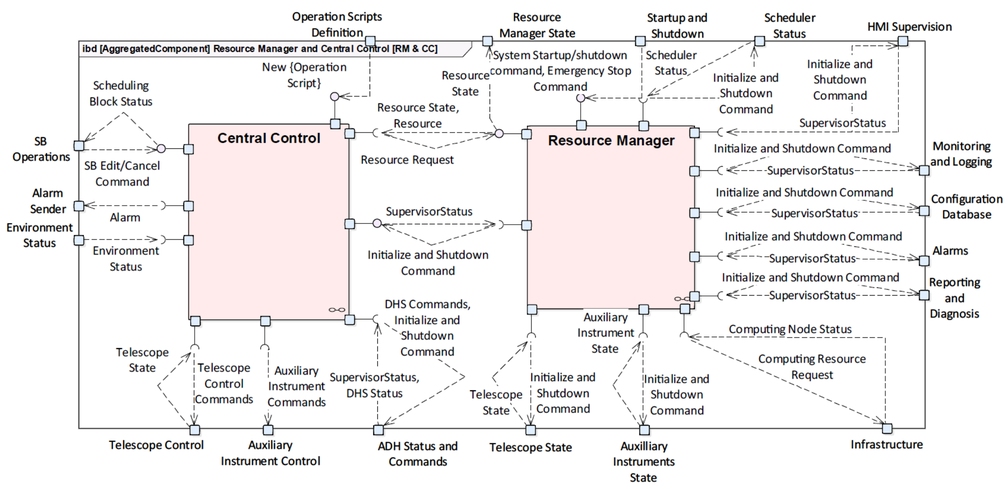
The Cherenkov Telescope Array (CTA) is the next-generation atmospheric Cherenkov gamma-ray observatory [1]. CTA will consist of two large arrays with 118 Cherenkov telescopes in total, deployed in the Paranal (Chile) and Roque de Los Muchachos (Canary Islands, Spain) Observatories.The Array Control and Data Acquisition (ACADA) [2, 3] system provides the means to execute observations and to handle the acquisition of scientific data in CTA. The Resource Manager & Central Control (RM&CC) sub-system is a core element of the ACADA system that implements the execution of observation requests received from the scheduler sub-system.

THE ACADA SYSTEM

ACADA will be implemented as distributed software system using the ALMA Common Software (ACS) [4], which is a set of application frameworks built on top of CORBA. ACS is based on a container-component model and supports the programming languages C++, Java and Python. ACADA is composed of several closely interrelated sub-systems (Short-term Scheduler, Transient Handler, Resource Manager and Central Control, Array Data Handler, Human Machine Interface, Science Alert Generation Pipeline, Array Alarm, Configuration, Reporting, Monitoring and Logging). Each CTA site will contain one instance of the ACADA system.

THE RESOURCE MANAGER AND CENTRAL CONTROL SYSTEM

RM&CC is one of the top-level sub-systems of ACADA. It was prototyped following the Model-Driven Architecture (MDA) approach of ACADA [5]. RM&CC comprises two components; namely, *Resource Manager* and *Central Control*.



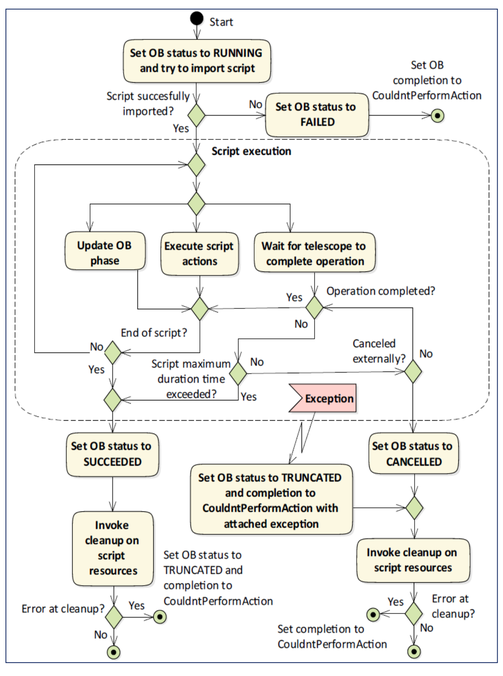
Logical view of RM&CC, representing the highest-level components and the most relevant data elements and interfaces of the system.

SCRIPTING ENVIRONMENT

The scripting environment provides the means to execute the main observatory operation modes. These are implemented as high-level Python scripts, and are executed by the Sub-array Sequencer component. This environment simplifies the script code by specifying a standard way of structuring the scripts and provides ways to test it.

```python
__phases__ = ["Configuring","Slewing","Tracking","Acquiring","Closing"]

def configuring():
    cameraConfig = schedulingBlock.config.camera_configuration
    telescopes.configureCameras(cameraConfig)

    # Allow Slewing phase to be run in parallel to this phase.
    allowPhaseStart("Slewing")
    # Do not end phase until all telescopes cameras are configured.
    telescopes.waitToConfigureCamera()

def slewing():
    # constructTarget is defined as a command
    resources.target = constructTarget()
    telescopes.startSlewing(resources.target)
    telescopes.waitToStopSlewing()

def tracking():
    trackingDuration = observationBlock.observing_conditions.duration
    telescopes.startTracking(trackingDuration,resources.target)

    startObservationTime(trackingDuration)
    allowPhaseStart("Acquiring")

    waitUntilObservationTimeFinished()
    telescopes.stopMotion()

def acquiring():
    daq().moveToNextOutputBlock(daqctrl.ZFITS_ZLIB)
    telescopes.startDataTaking()

    # Tracking and acquiring are running in parallel and both complete
    # after observation time is up.
    waitUntilObservationTimeFinished()
    daq().moveToNextOutputBlock(daqctrl.DISABLE)
```

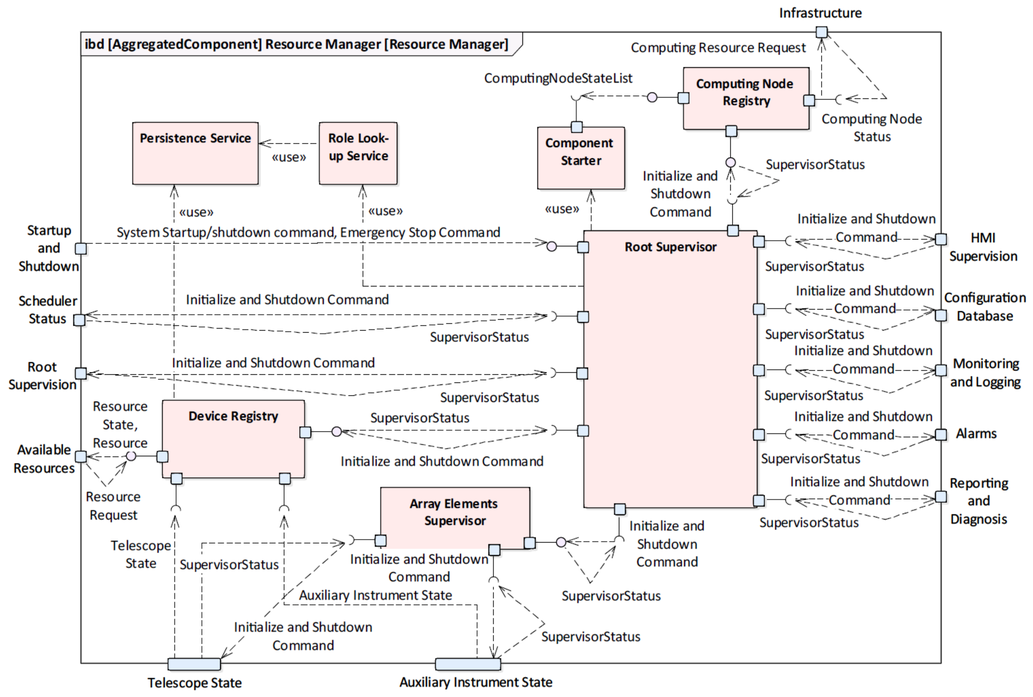
An example of a high-level Python script.



Script execution diagram.

THE RESOURCE MANAGER

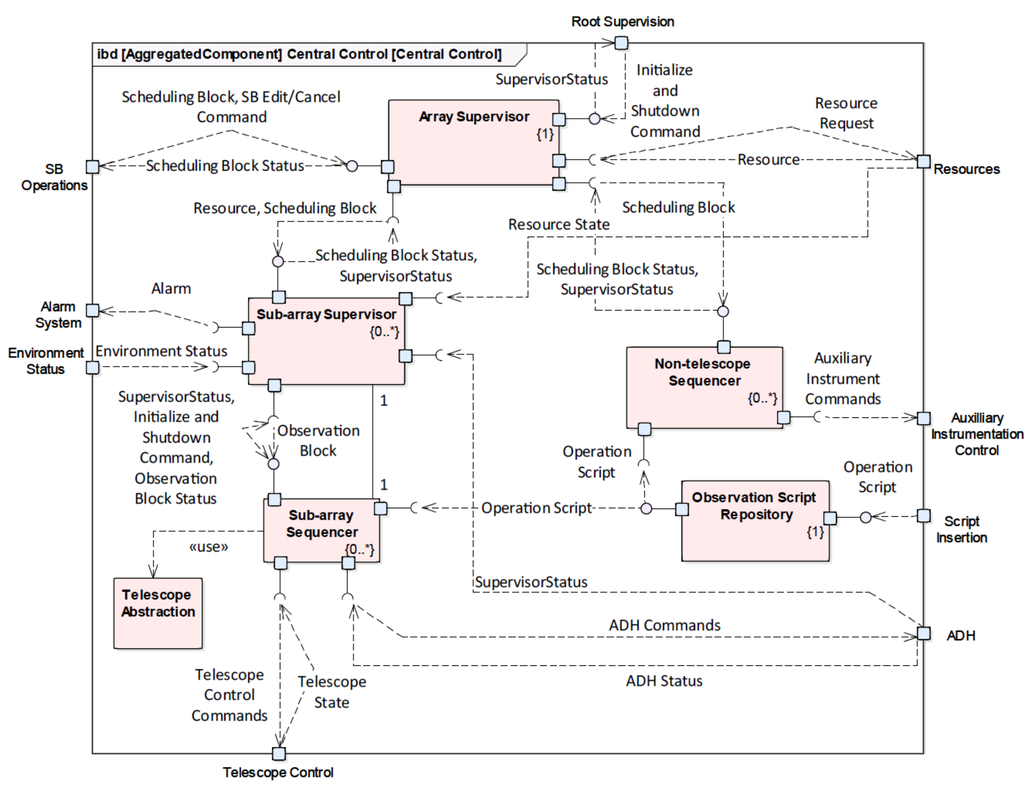
The Resource Manager provides to all ACADA sub-systems administrative and infrastructure services concerning various resources. The later comprise telescopes, auxiliary instruments, computing nodes and ACADA components.



Logical view of the Resource Manager component.

THE CENTRAL CONTROL

The Central Control component implements the execution of scheduling blocks received from the Scheduler sub-system.



Logical view of the Central Control component.

CONCLUSIONS

This prototype is intended as a proof of concept, applying a model driven approach to component based modelling of the ACADA system. The current version of the RM&CC system is capable of executing basic observation modes and running of multiple operations on various sub-arrays simultaneously. It also provides the main functionality of a supervision tree that includes dynamic instantiation, start, supervision, shutdown and replacement of a supervised component with a successor, in the case it vanished or reached an error state. RM&CC has been successfully integrated with the Scheduler and HMI sub-systems.

REFERENCES

[1] B. S. Acharya *et al,* “Introducing the CTA concept” Astroparticle Physics, vol 43, pp. 3-18, 2013.

[2] I. Oya *et al,* “The Array Control and Data Acquisition System of the Cherenkov Telescope Array”, presented at ICALEPCS’19, October 2019, paper WEMPR05, this conference.

[3] E. Antolini *et al,* “Quality Assurance of SCADA System of the Cherenkov Telescope Array Observatory”, presented at ICALEPCS’19, October 2019, this conference.

[4] G. Chiozzi *et al,* “CORBA-based Common Software for the ALMA project”, in Proc. SPIE 4848, 43, 2002.

[5] I. Oya *et al,* “The software architecture to control the Cherenkov Telescope Array”, in *Proc.* SPIE 9913, Software and Cyberinfrastructure for Astronomy IV, paper 991303, 2016.

[1]DESY, Zeuthen, Germany, [2]CTAO gGmbH, Heidelberg, Germany, [3]Cosylab d.d., Slovenia, [4]Humboldt-Universitaet zu Berlin, Germany, [5]INAF – Osservatorio Astronomico di Brera, Milan, Italy, [6]https://www.cta-observatory.org