Artificial Intelligence for the CTA Observatory Scheduler

J. Colomé[1]{colome@ice.cat}, P. Colomer[2], J. Campreciós[1], T. Coiffard[2]
[1]Institut de Ciències de l'Espai (IEEC – CSIC), Barcelona, Spain
[2]GTD Sistemas de Información, Barcelona, Spain

Introduction

* http://cta-observatory.org

The Cherenkov Telescope Array (CTA*) project will be the next generation ground-based very high energy gamma-ray instrument. The CTA Consortium is composed by more than 1000 scientists and engineers, from 186 different institutes located in 28 countries. The project is planned to start the operation phase in 2018.

CTA will consist of two arrays of Cherenkov telescopes operated as a proposal-driven open observatory. It will handle several observation modes and will have to operate tens of telescopes with a highly efficient and reliable control.

The operation, control and readout of a large distributed multi-telescope array leads to new challenges, which have to be met to guarantee the required overall CTA performance. In contrast to existing IACT projects like MAGIC, VERITAS and H.E.S.S., CTA is characterized by an increased complexity of the system and higher autonomy and reliability expected from the individual telescopes. Thus, CTA is creating new demands on the Array Control and Data Acquisition system (ACTL). And, in particular, the CTA planning tool (scheduler) is a key element in the control layer for the optimization of the observatory time.

Figure 1. MAGIC telescopes in La Palma (ORM, Spain) & HESS telescopes in Namibia

The main purpose of the scheduler for CTA is the allocation of multiple tasks to one single array or to multiple sub-arrays of telescopes, while maximizing the scientific return of the facility and minimizing the operational costs. The scheduler considers long- and short-term varying conditions to optimize the prioritization of tasks.

The Institute for Space Sciences (IEEC-CSIC) has developed a scheduler prototype, in collaboration with GTD S.I. Company, and leads the task to develop such an important tool to achieve an efficient operation of this large infrastructure.

The scheduler for CTA is based on Artificial Intelligence techniques. It uses Guarded Discrete Stochastic Neural Network and Constraint Propagation to obtain different long-term solutions from the search space. Afterwards, the most efficient solution is selected to be processed by the short-term scheduler, which is based on an Ant Colony Optimization algorithm that re-schedules the tasks depending on different notifications (i.e., change on weather conditions) received from the array control system.

Proposal Based on Artificial Intelligence

The Scheduler for CTA is based on an algorithm composed by a GDS neural network & using Constraint Satisfaction Problem (CSP) for constraint propagation. An Ant Colony Optimization (ACO) algorithm is also used to avoid hard constraint violations. The benefits of this algorithm are:

➔ Scheduling conditions can be changed during execution without the need to start from scratch

➔ It may optimize an entire night, week, month, etc., for all kind of time and resource constraints and remove conflicts

➔ Optimize more than one objective with no lost of information

Scheduling process

➔ Initialization phase: Objectives to optimize and specification of constraints and priorities & System configuration (Observatory, proposals, period to schedule, etc.)

➔ Long-term planning for a full period: Recomputed every morning

➔ Long-term planning: Input for the short-term

➔ Most optimized schedule is returned: Tasks (SBs) + Execution time

➔ Short-term planning: Schedule updated based on system events

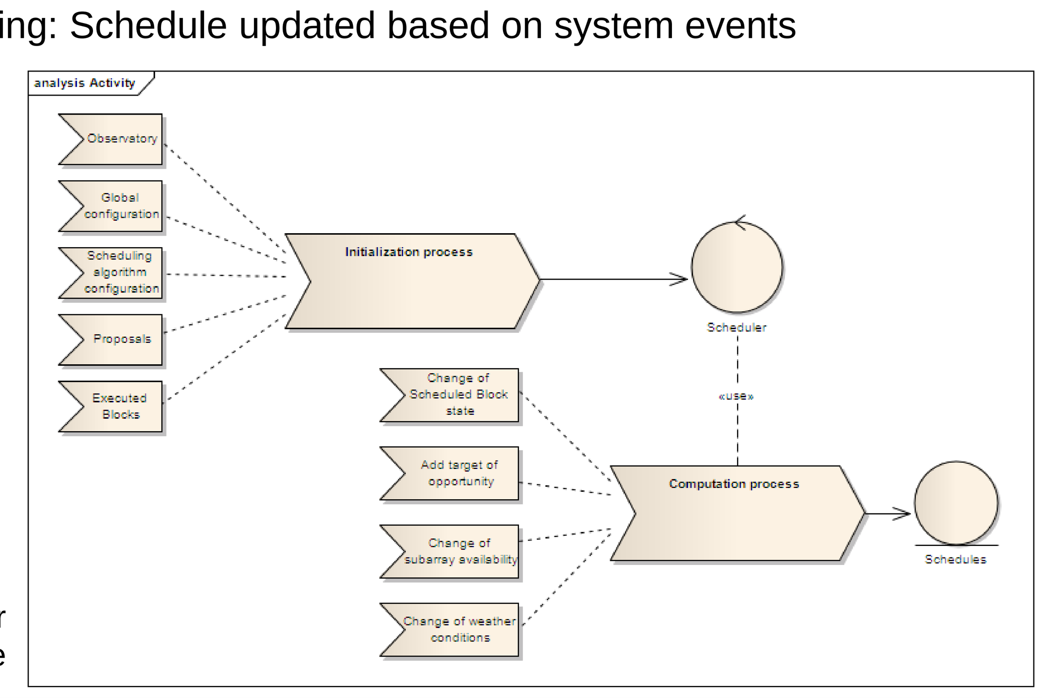


Figure 2. Scheduler algorithm cycle

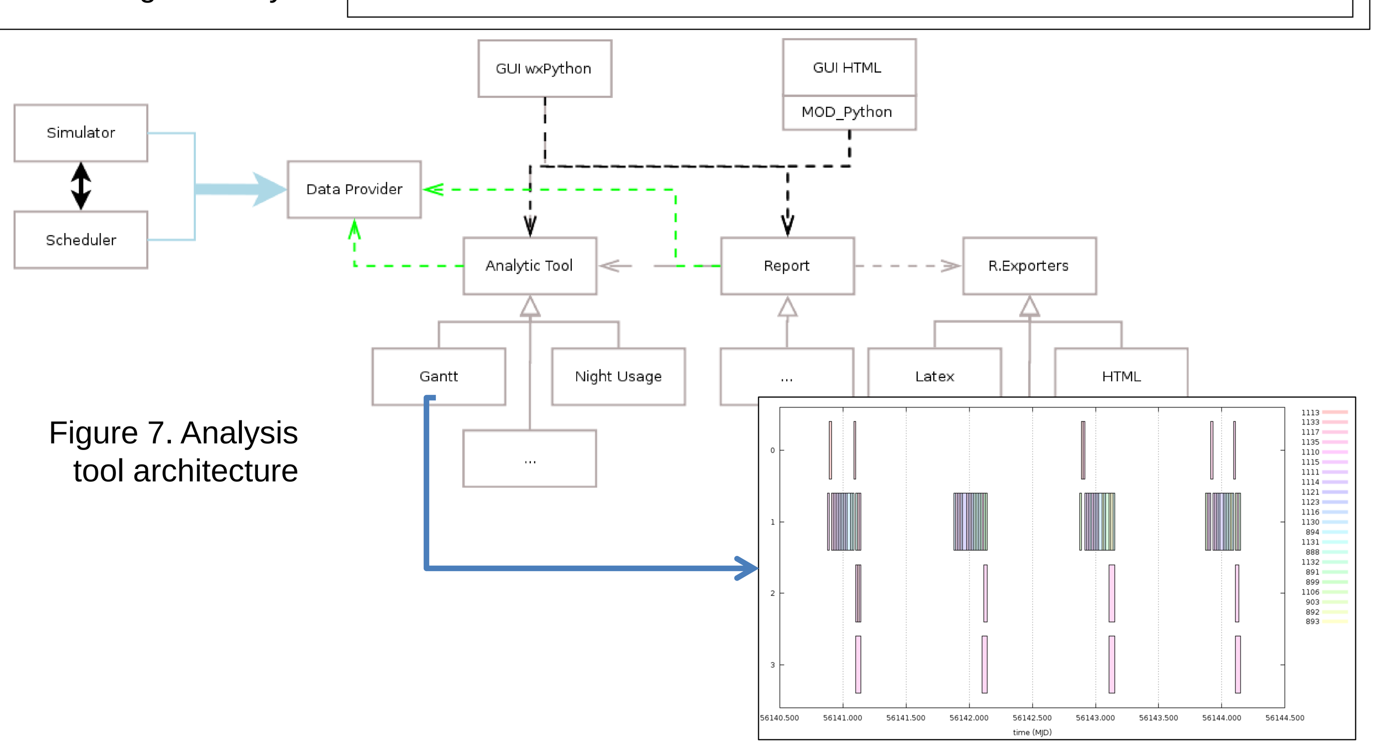


Figure 7. Analysis tool architecture

Planning Conditions of the CTA Observatory

Observatory control layer

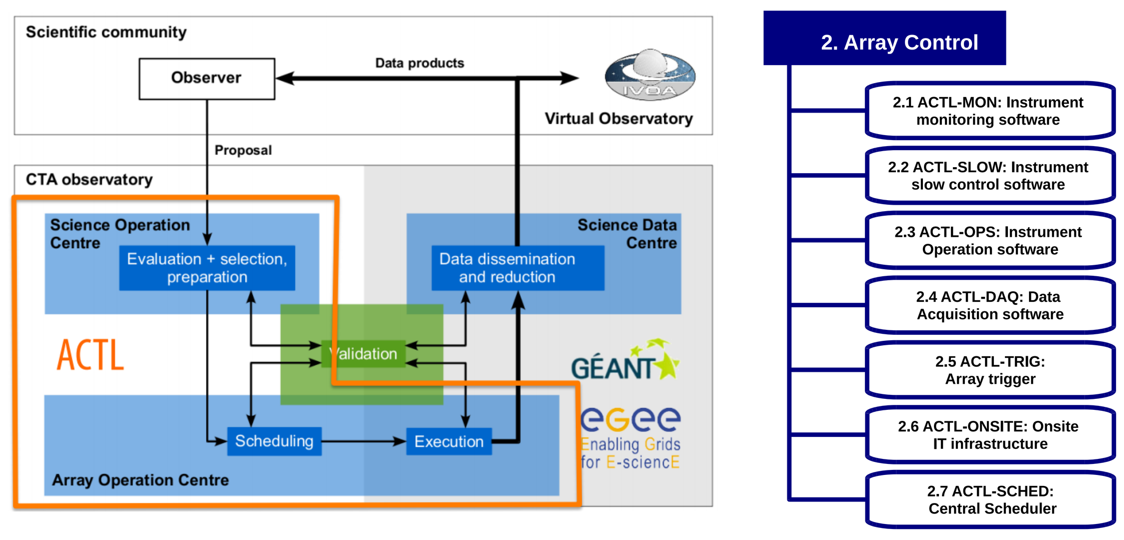


Figure 3. Scheduler in the ACTL work package

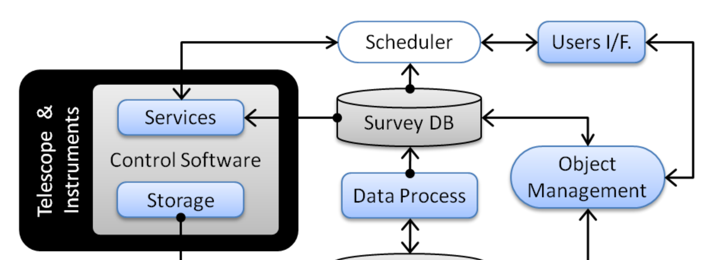


Figure 4. Scheduler in the observatory control architecture

The scheduler for CTA is included in the array control work package (see fig. 3), which is devoted to build the control layer of the observatory. This layer is responsible to define different operative features that are relevant for the subsequent definition of the planning tool:

➔ User access profiles

➔ Operation tasks (Science & Calibration)

➔ Observation modes

➔ Observing time distribution or Scheduling Blocks (SB)

Constraints	Long-term	Mid-term	Short-term
Target visibility			
Remaining time for observation			
Avoidance areas			
Tasks dependences			
Maximum number of executions			
Time constraints & Monitoring			
Calibration required			
System conditions & configuration			
Environment conditions			

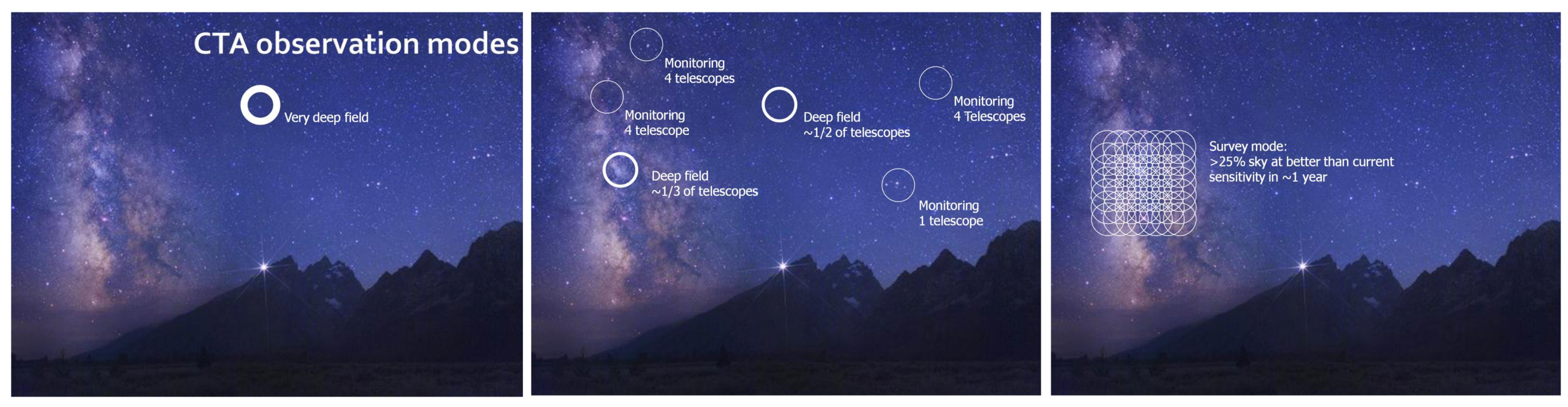


Figure 5. CTA observation modes

Scheduling cycles

➔ Short-term: Adapts the selected task to the varying conditions in almost real-time.

➔ Mid-term: Considers intermediate time-scale situations and may carry out a pre-selection of tasks to improve the time response given by the short-term scheduler.

➔ Long-term: Provides season planning (months) to optimize the subsequent short-term selection.

Problem characterization

➔ Objectives

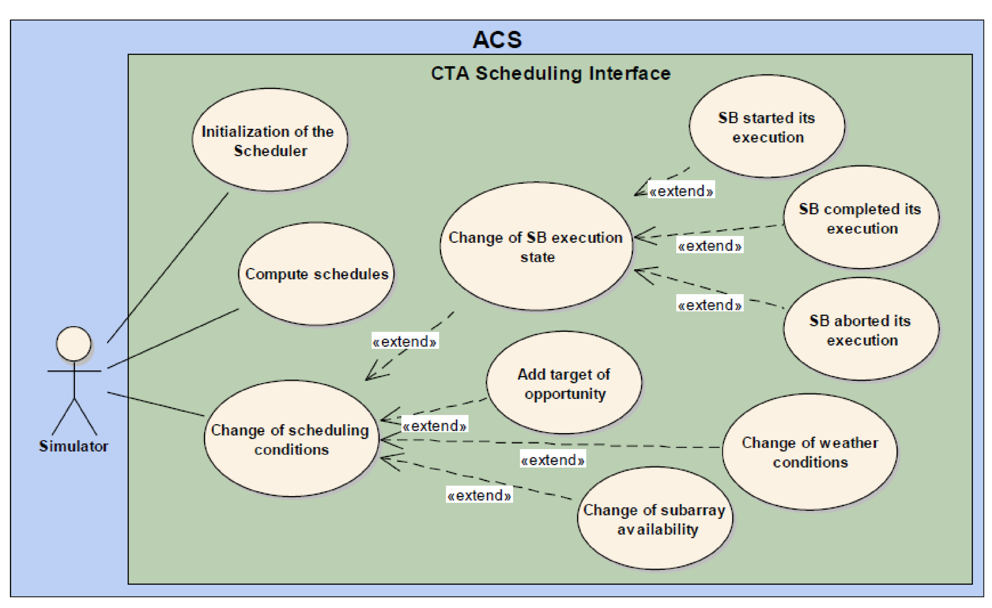
- Optimization of scientific return: Promote observation of the priority targets & completion of proposals.
- Optimization of resource: Maximize the time allocated to scientific tasks & minimize time overheads (i.e., slewing)

➔ **Soft constraints** (conditions that should be satisfied)

➔ **Hard constraints** (conditions that must be satisfied): target visibility, location, time, dependencies, number of observations, system status, environment conditions, etc.

Design of the Planning Tool

Use case model & Software Architecture



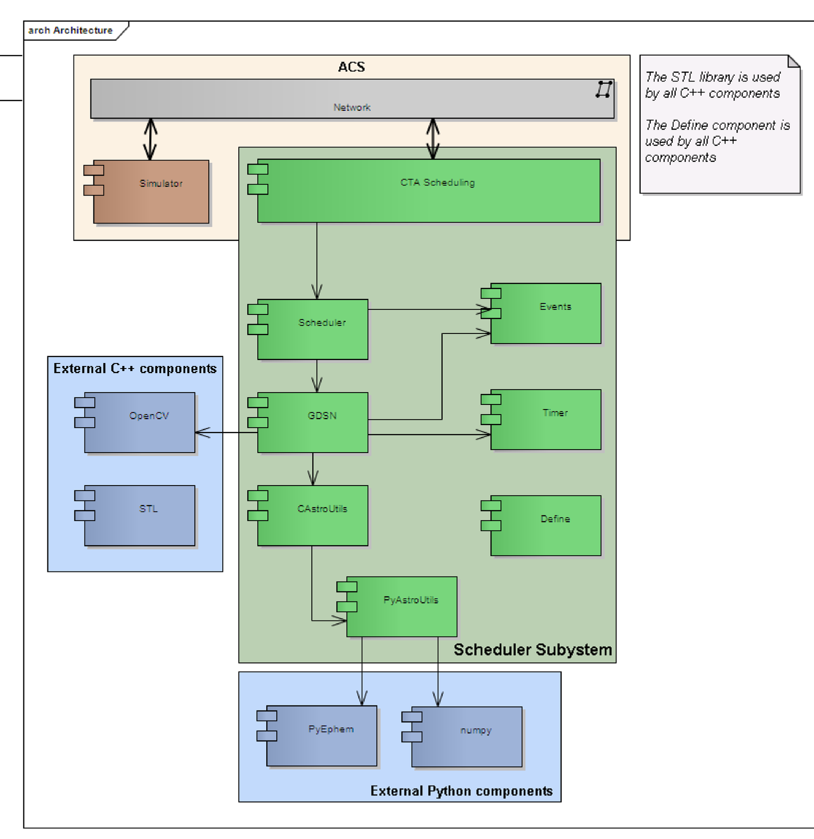


Figure 6. CTA Scheduler architecture

This modular architecture (fig.6) is designed to communicate with the CTA array control system using the ALMA Common Software (ACS) infrastructure. A simulator is also developed to evaluate the performance of the planning tool and communicates with the scheduler using the same infrastructure.

The scheduler component defines all functionalities of the Scheduling Subsystem. It also contains abstract classes needed to build new scheduling algorithms. These abstract classes define the behavior of all scheduling algorithms and provides an easy exchange of the algorithm in use.

GDSNScheduler implements two long-term and two short-term scheduling algorithms. All these algorithms are based on a Guarded Discrete Stochastic Neural Network. One of the short-term algorithms extends the GDSN by merging it with an Ant Colony Optimization (ACO) algorithm. Both long-term algorithms use the same GDSN but with different constraints.

Analysis & Conclusions

An analysis tool is under development (see architecture in figure 7). It is devoted to help comparing the statistics for the different scenarios and using predefined evaluation metrics.

Current prototype was developed after an exhaustive analysis of the existing algorithms (Colomé et al., 2012, Proceedings SPIE, "Research on schedulers for astronomical obser.") that helped selecting the best solution for the CTA project. This was identified as a Flexible Job Shop problem that could be solved with the described algorithm. Initial tests prove the good performance of the proposed algorithm. The planning tool prototype is designed for an easy integration in the CTA control layer.

Further Work

Development of the Operative Version of the software. Implementation of a visualization tool for interacting with the planning tool. Analyze how to generalize the planning tool for other missions.

Acknowledgments

We gratefully acknowledge financial support from the Spanish Ministry of Economy and Competitiveness (MINECO), the Centre for the Development of Industrial Technology (CDTI), the "Fondo Europeo de Desarrollo Regional" (FEDER) and the support from the agencies and organizations listed in this page: http://www.cta-observatory.org/?q=node/22

Poster available at:
http://goo.gl/M1v96v