The LSST OCS scheduler design

Francisco Delgado*[a], Germán Schumacher[a]
[a]Cerro Tololo Inter-American Observatory, Casilla 603 La Serena CHILE

ABSTRACT

The Large Synoptic Survey Telescope (LSST) is a complex system of systems with demanding performance and operational requirements. The nature of its scientific goals requires a special Observatory Control System (OCS) and particularly a very specialized automatic Scheduler. The OCS Scheduler is an autonomous software component that drives the survey, selecting the detailed sequence of visits in real time, taking into account multiple science programs, the current external and internal conditions, and the history of observations. We have developed a SysML model for the OCS Scheduler that fits coherently in the OCS and LSST integrated model. We have also developed a prototype of the Scheduler that implements the scheduling algorithms in the simulation environment provided by the Operations Simulator, where the environment and the observatory are modeled with real weather data and detailed kinematics parameters. This paper expands on the Scheduler architecture and the proposed algorithms to achieve the survey goals.

Keywords: LSST, scheduling, SysML, OCS

1. INTRODUCTION

The Large Synoptic Survey Telescope (LSST) is envisioned to be a large, wide-field ground based telescope designed to obtain sequential images covering the entire visible sky every few nights. By its very nature, the LSST will be a highly "robotic" telescope, performing a series of observations following a regular cadence, to optimize sky coverage and data throughput. With its unprecedented combination of collecting area and field of view, the LSST will image large areas of the sky frequently and with great sensitivity. The cadence of these observations, the order in which different fields of view are observed in each filter and the frequency with which they are revisited, will determine just how much sky will be covered, for how many visits, and with what temporal sampling. To accomplish those goals, the LSST is conceived as a system of systems composed of a Telescope, a Camera and a Data Management System, that must perform as a fully integrated unit.

Given the cadence and other functional requirements of the survey, the notion of a coordinating entity dealing with many aspects of a "robotic" telescope, is readily grasped. In that context, the Observatory Control System (OCS) is the master control system that schedules, commands, coordinates, and monitors the observatory. The OCS is responsible for high level observatory operations including user interfaces, sequencing, resource allocation and system monitoring and maintenance. The OCS orchestrates and controls all aspects of the observatory for all observations (science, calibration, and engineering) in all of the operational modes.

One of the most important and complex functional requirements for the OCS is the conduction of the survey. The component that is allocated with most of the activities associated with the survey is the OCS Scheduler. The Scheduler is the engine that automatically decides the sequence of targets to observe during the night. To achieve that goal, the Scheduler runs a set of science programs to analyze the possible targets following the different science objectives defined for the survey. Each candidate target is ranked according to the progress of each science program recorded in an internal observation database. The score of each candidate considers time sampling, color sampling, sky coverage and depth in number of visits. The observing conditions are also taken into account, such as airmass, seeing and sky brightness. Once all the targets have received a science priority score, the scheduler computes the cost of slewing the observatory from the current position to select the best score/cost combination and proceed with the observation. This process is executed constantly to adapt the scheduling to the always changing environment conditions and obtain an optimized survey operation.

*fdelgado@ctio.noao.edu; phone +56-51-2205303 +1-520-3188000; lsst.org

Observatory Operations: Strategies, Processes, and Systems V, edited by Alison B. Peck, Chris R. Benn, Robert L. Seaman, Proc. of SPIE Vol. 9080, 908000 © 2014 SPIE
CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2056871

2. SYSTEM ARCHITECTURE

The LSST Control System driving the LSST mission, is being designed in terms of a distributed architecture, comprised of many processors and devices making an interconnected network. The Control Software is tailored to efficiently and safely perform astronomical observations individually or through automated scheduling. It also provides support for engineering, set-up and maintenance, and creates a dynamic environment for development and evolution of control applications. The Control Software is developed to implement a hierarchical distributed architecture to maximize the control efficiency and to support the highly robotic nature of the LSST System.

Besides control and monitoring, another requirement for the Control System is to perform extensive data telemetry capture on every aspect of the system operation and system actions. This includes message exchanges and parameters associated with the observing environment. The telemetry should be captured in real-time and organized in permanent database storage, to provide science metadata and to support troubleshooting, maintenance and analysis activities. This information is analyzed daily in order to perform preventive maintenance, to optimize the system, and to measure performance trends over long time periods. Telemetry is recorded throughout the regular daily cadence of observatory operations, including nightly data acquisition, engineering, calibration and science.

A communications layer that provides an efficient and reliable way of exchanging messages between the different subsystems and components has been considered. The messages consist of control information that needs to be delivered in a predictable fashion, and contain also of status information utilized to monitor system performance. In the case of the LSST system, a requirement for extensive telemetry capture adds more demand on that communications layer.

The communications layer is based on a data-centric architecture for information distribution, consisting of nodes that communicate simply by sending the data they have and asking for the data they need. Publish-subscribe nodes "subscribe" to data they need and "publish" information they produce. Publish-subscribe adds a data model to messaging. The fundamental communications model provides both discovery (what data should be sent) and delivery (when and where to send it). Publish-Subscribe middleware is the key enabling technology for data-centric design. The data model means one can essentially ignore the complexity of the data flow; each node gets the data it needs from the bus.

A fundamental precept of this architecture is that all software components are individually accessible from any other component unless access is explicitly restricted. The extent to which this accessibility is organized (hierarchically or otherwise) and how access is restricted are properties of the system design, not the infrastructure. This design decouples the dataflow, as a mean to contain complexity. Decomposition, modularization and separation of concerns are typical techniques. However, the level of decoupling between components of the system can vary; loosely-coupled and tightly-coupled systems. A well decomposed system shall be loosely-coupled. This is important not only to contain complexity, but also to increase scalability. A system with decoupled components allows replacing one component without impacting the rest of the system.

3. OCS ARCHITECTURE

The Observatory Control System (OCS) is the master control system that schedules, commands, coordinates, and monitors the observatory. The OCS is responsible for high level observatory operations including user interfaces, sequencing, resource allocation and system monitoring and maintenance. The OCS orchestrates and controls all aspects of the observatory for all observations (science, calibration, and engineering) and all operation modes. The OCS coordinates the camera, telescope and data management subsystems for an integrated operation during the survey. Through the OCS the system can be started, monitored, adjusted during operations and stopped, both locally and remotely. The OCS provides the means to support safe observatory operations during day and night.

The OCS is the hierarchical master for the survey operation of the observatory, with interfaces to the 3 subsystems. The OCS interacts with the respective logical master of each subsystem using high level commands; this subsystem master is in charge of controlling its internal devices. In this way the OCS can control and monitor the subsystems in a very clean and decoupled way, without the need of accessing the internal devices of the controlled subsystem.

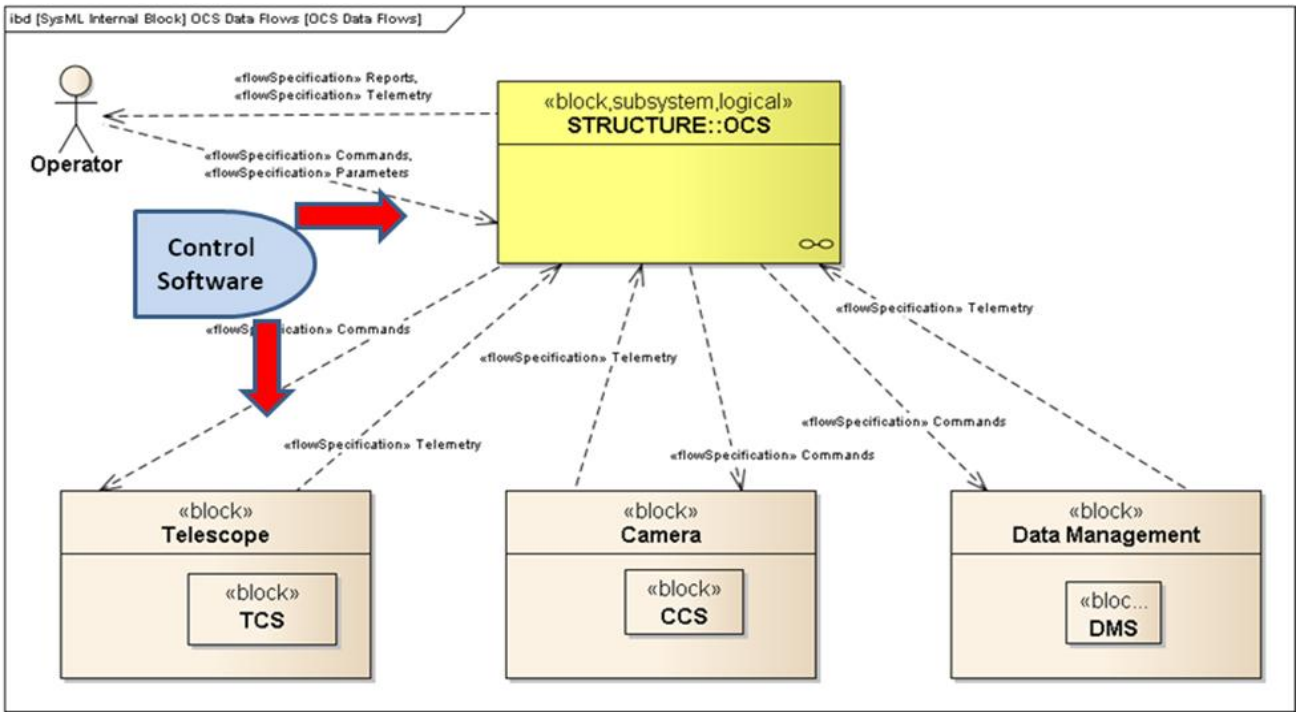

Figure 1. Hierarchy of OCS and the other subsystems, from a control stand of point.

From the logical point of view of the commands, there is no direct interaction between the 3 subsystems Telescope, Camera and Data Management. The OCS is the only subsystem that commands them in normal operations. Similarly, the telemetry is transmitted through the communications middleware, a service considered as part of the OCS, so each subsystem has a telemetry interface defined only to the OCS.

The internal architecture of the OCS follows the same paradigm of the LSST control system. The OCS components are distributed and exchange commands, events and telemetry on the same communications middleware implementation. This is, each OCS component publish and subscribe to its relevant data, building a highly decoupled subsystem in which the actors cooperate for the general behavior.

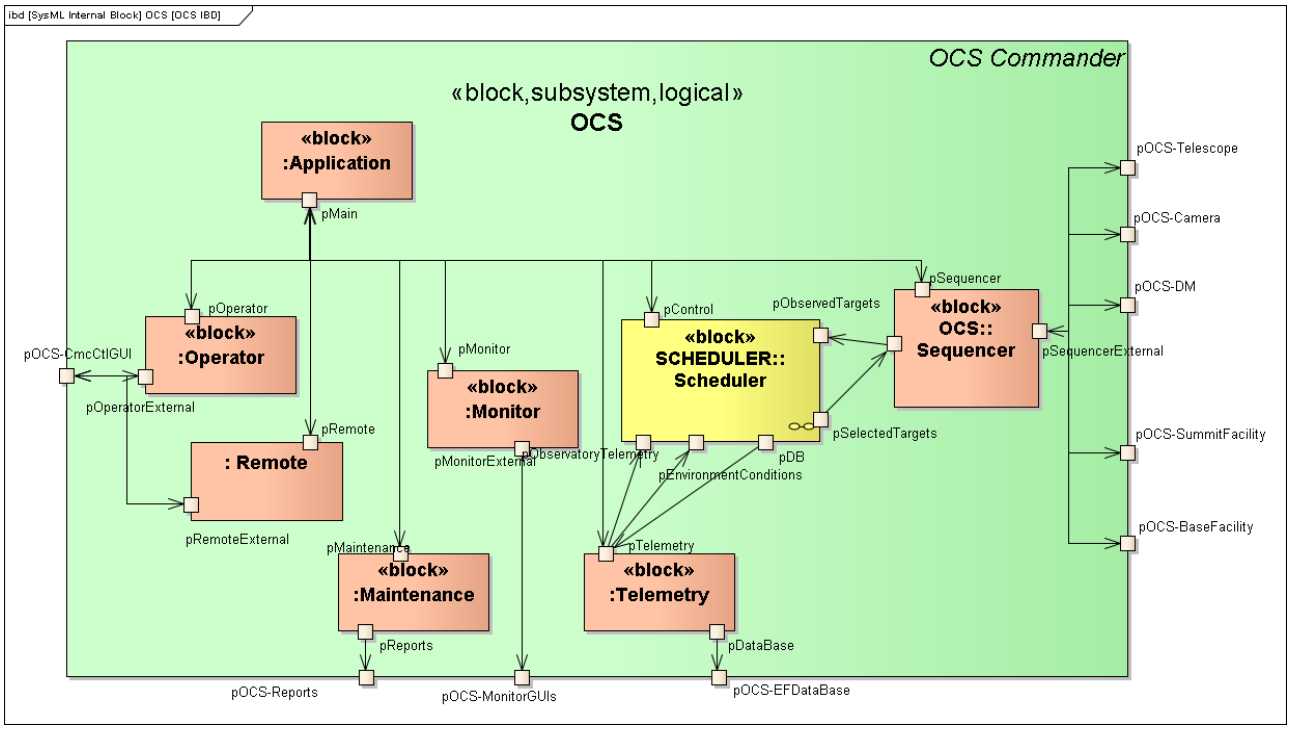

Figure 2. Internal block diagram of OCS, showing the basic data flows and interface ports.

4. SCHEDULER CONCEPT

The scheduling function of the OCS, considered the key component of this subsystem, has as its main feature the ability to handle demands from several competing science programs, with different requirements on cadence and sky location observations, and produces a dynamic schedule that optimizes the observation time.

The OCS Scheduler is a very complex component and as such, it has being prototyped in the project Operations Simulator (OpSim) and the effort to improve it continues. In OpSim, a prototype of the scheduler is implemented in a simulated environment that includes a telescope/camera model, weather model with actual historical data, an astronomical sky model and a database for registering the simulated survey. It has plenty of parameters and flexibility for experimenting with different cadences and science goals in order to verify that the design and the strategy comply with the requirements, reducing the risk on this component in a significant measure.

In the early stage of the project the decision to prototype a scheduler in a simulation environment gave the opportunity to experiment with different approaches. The main concept that prevailed was that multiple science programs can compete for the telescope time, offering multiple ranked targets as candidates. Each science program ranks their targets of interest according to its own history and goals. This way the scheduler counts internally with multiple options, each one with its own science value for the given moment. Then, a cost is calculated for each one and an overall score is computed. Following this score, the best target is selected. This simple generic algorithm demonstrated to be effective enough and scalable in nature. Also, despite the fact that the basic rules are very simple, the high number of elements involved in this interaction have shown to produce an emergent behavior that solves the problem sometimes in interesting and unexpected ways.

Combining this practical experience in the Operations Simulator, plus the requirements flowed down from the SRD, OSS and OCS, a number of guidelines were established to produce the Scheduler design.

Fully automatic operation

One of the requirements for the Scheduler is a fully automatic operation for the main survey mode. The Scheduler selects the targets in an automatic fashion, utilizing the abundant telemetry of the observatory and the environment, and the configured programs to pursue.

Dynamic adaptation to environment and observatory conditions

The Scheduler needs to be adaptive in order to get the most scientific value from the changing external conditions. Environment telemetry, like seeing and clouds, are inputs considered for selecting the available targets at every moment according to the specific and different constraints from the science programs running. The observation schedule has to be evaluated constantly to adapt.

The observatory conditions can also change, like the speed of one of the mechanical elements contributing to the slew. Knowing in advance the slew cost of a visit is important to compare the alternatives, and the scheduler needs to account for any important change in that cost structure.

Sky field map, tiling regions

In order to make the scheduling problem manageable, the sky is divided in fields using a tiling map. Each field is assigned with an identifier, so a target is a field/filter combination.

Fully configurable set of concurrent competing science programs

Multiple science programs can be configured for the survey. Each one needs enough parameters to select the sky region of interest, the filters, sky brightness and airmass limits, cadence definition, sequences, ranking algorithms, etc. The science programs shall be fully shapeable through configuration files. Some science programs optimize visits distribution. Others optimize visits interval distribution. Some others push for time interval/filter sequences. Scripted visit sequences are also possible on a science program.

Target score balances science value and time cost

Each selected target has to come out from a scoring algorithm that combines the value ranked by the science programs proposing the targets, and the cost of taking that target given the initial configuration of the observatory.

Sky brightness dynamically modeled for each sky field

One of the parameters for ranking targets is the expected sky brightness of the field. Telemetry will be available for this purpose, but for look-ahead purposes and field map precision, a sky brightness model is also needed internally in the scheduler.

Comprehensive observatory kinematic model for slew time optimizations

The cost function for scoring the targets is basically the slew delay from the current telescope configuration. A detailed telescope model is included to estimate this slew time considering all the components of the movement and activities for a visit:

- mount azimuth with cable wrap limits
- mount altitude
- mount settle time
- rotator with cable wrap limits
- filter change
- shutter open and close movement
- detectors integration time
- camera read-out
- dome azimuth

- dome shutter
- active optics

Observations database

A detailed observations database is needed by the scheduler to properly rank future visits. The information has to include not only the telescope pointings and filters used, sky brightness, detailed slew movements and times, airmass, actual seeing, transparency, rank value, science programs involved, sequencing data, etc. Everything that may be needed to evaluate the progress of the survey and rank the future schedule.

This is an internal database to the scheduler, not necessarily dependant on the observatory engineering facility database.

Look-ahead

To optimize the observations distribution, improve uniformity and achieve efficiency in sequences, not only the past history and present conditions are needed. Some information about the known future is important as well, like the pre-calculated visibility and observation parameters of the targets in some defined time-window. The sky brightness model, the telescope model, sun and moon, are some examples of parameters that can be pre-calculated to anticipate observing conditions.

Scheduling parameters can be fine tuned to direct the survey

Besides the science programs parameters, the scheduler shall include higher level parameters for adjusting the survey, like relative priorities, limits across the board, behavior of scheduling algorithms, optimization values, etc.

5. SCHEDULER MODEL-BASED SYSTEMS ENGINEERING

Systems development can be seen as a process of transforming a set of requirements into a real system that provides a solution to a known problem. To carry out this process in the most effective and efficient way, an organization needs to ensure that anything built as part of the system, in either hardware or software, directly relates to a defined requirement. This activity ranges from tracing derived requirements, over satisfaction of requirements by design elements to verification of requirements by test cases.

Requirements traceability deals with controlling this process, and ensuring that a system is not 'over-engineered' above its requirement needs or 'under-engineered' by missing requirements. Tracing requirements sounds like a simple and obvious element of the systems engineering process, but in practice, in dynamic, large scale and complex systems, it becomes a major problem. The amount of design information and deliverables created within the project can be massive, and requirements can be lost in the morass of information available. Requirements also change, and the effects of these changes need to be monitored and controlled.

To support the development of the LSST system, the Observatory Control System and the Scheduler in particular, the OMG System Modeling Language (SysML) has been selected as the framework for the systems engineering analysis and documentation. Using this framework, models for the overall system architecture and different observatory subsystems have been built, describing and relating requirements, structure, interfaces and behavior.

The Scheduler design begins with the requirements analysis of the system. All the OCS requirements are located in the main SysML model and the Observatory Control System requirements document is generated directly from that model. The Model-based Systems Engineering methodology has been followed to produce the analysis of the Scheduler software, all interrelated in analysis and validation diagrams.

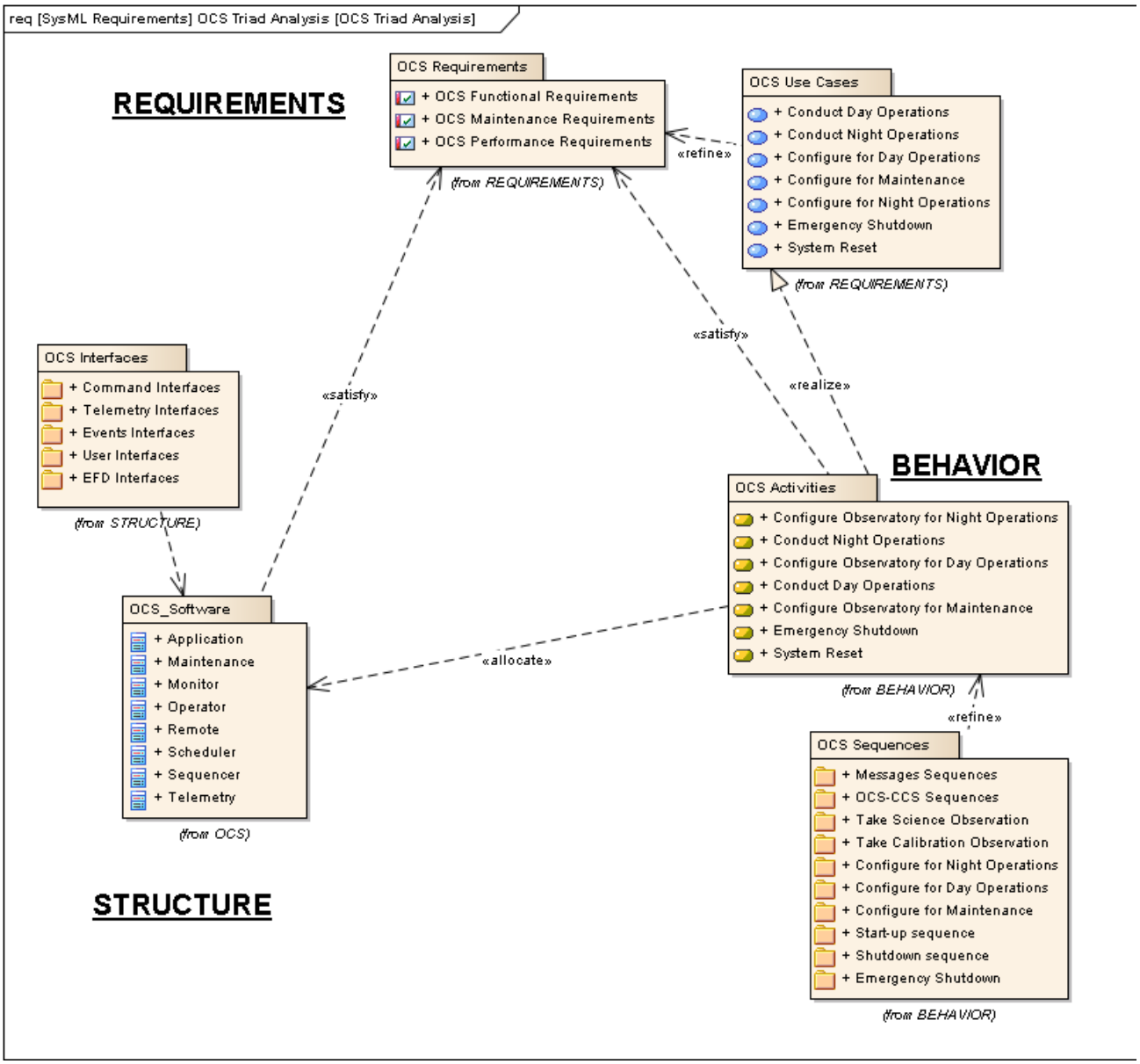

Figure 3. Hierarchy of OCS and the other subsystems, from a control stand of point.

An important aspect has been the adoption of a methodology of model validation, in terms of a triad combining requirements, structure and behavior: triangulation of the model to ensure architectural integrity, align functional requirements with Use Case counterparts, satisfy and verify requirements with appropriate structures and behaviors. This ensures that all requirements, including derived requirements, are satisfied by at least one structural modeling element and at least one behavioral modeling element.

REQUIREMENTS

The Scheduler requirements are derived after a flow-down process from the Science Requirements Document, the Observatory System Specification and the Observatory Control System Requirements. Specifically, to respond to the need for “a universal cadence that would result in a common database of observations to be used by all science

programs". The Scheduler shall provide an observing program consisting of a sequence of telescope pointings and data acquisition cycles suitable for automated operations. The observing program will be optimized for key science, general science use, and observing efficiency, according to rules or scoring defined and approved by LSST science management and implemented under the supervision of the Schedule Scientist.

There is also the dependency from the scientific community through the prototyping effort in the Operations Simulator, putting the required skills in the Scheduler to the test utilizing different science cases.

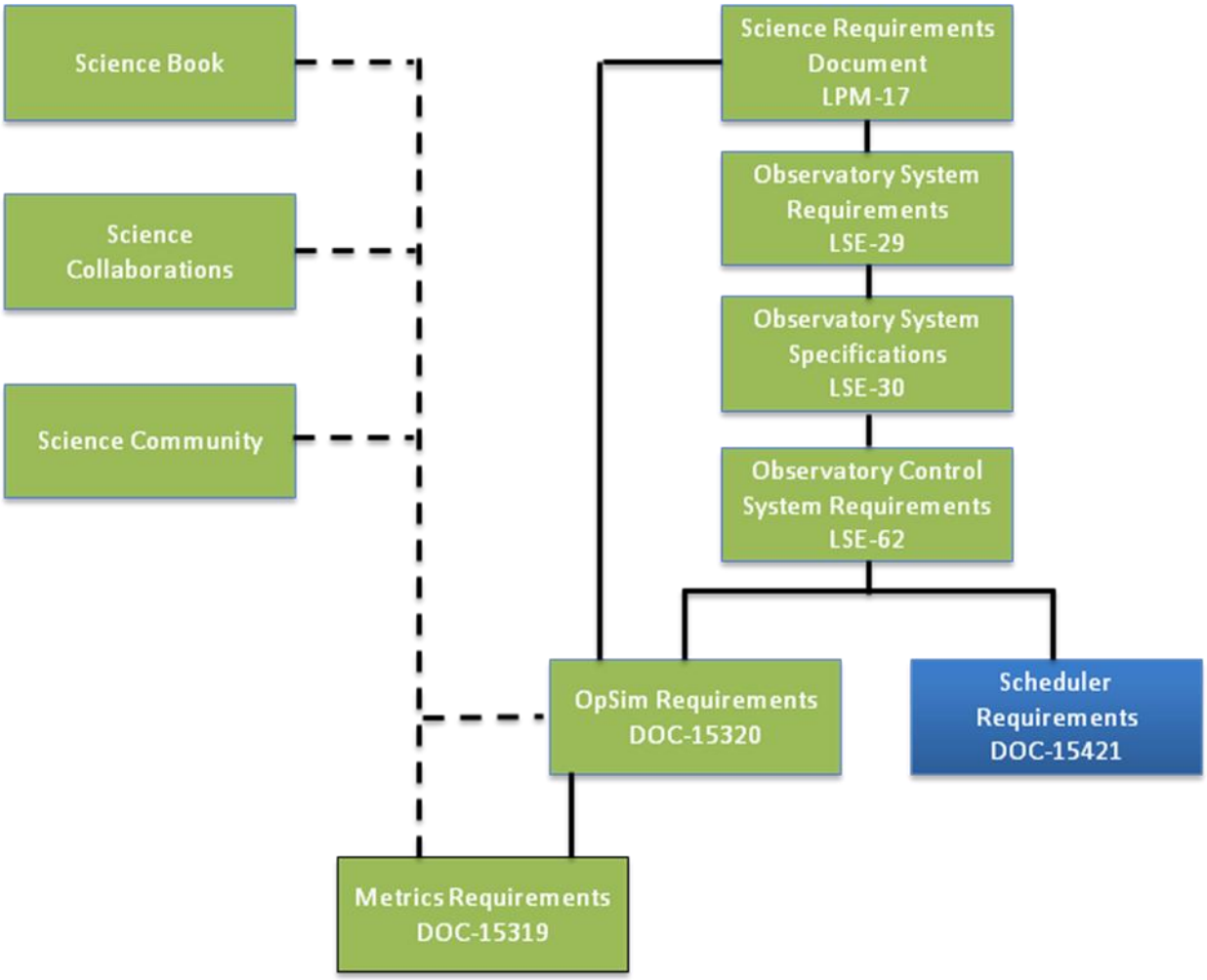

Figure 4. Requirements flow-down for the Scheduler.

Each one of the first level Scheduler requirements is properly traced to its respective requirement in the Observatory System specifications, illustrated in the SysML traceability diagram.

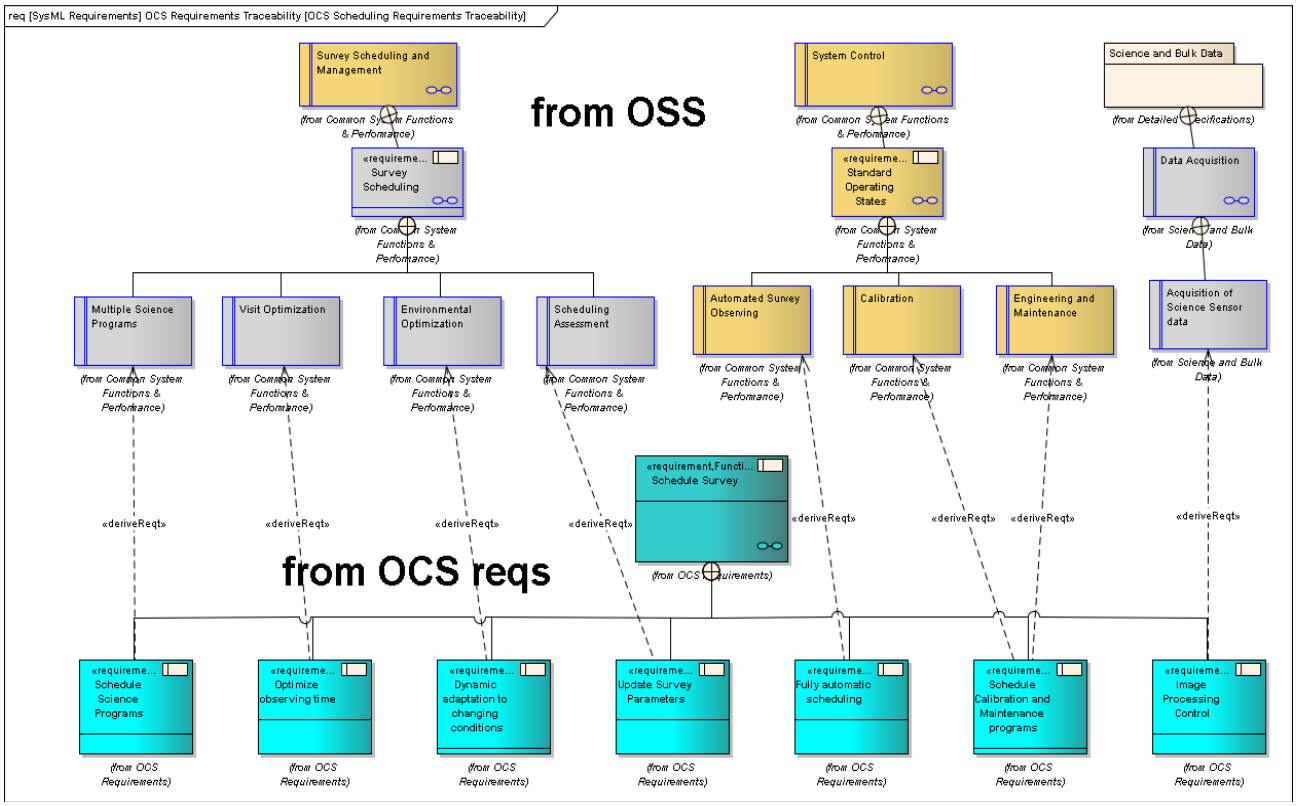

Figure 5. Scheduler requirements traceability to Observatory System Specifications.

STRUCTURE

This is the block definition diagram for the OCS Scheduler. Each component is an actor in the internal Scheduler architecture, and they cooperate in the operations publishing and subscribing to the internal messages in the same paradigm as the overall LSST system.

MBSE has been applied to the analysis of this structure with the requirements, to assure that each Scheduler requirement is satisfied by at least one Scheduler component, and each component is satisfying at least one requirement.

The Scheduler is the heart of the OCS that enables the automated survey conduction. Following the same paradigm from the OCS, the internal architecture of the Scheduler uses the same design style. All its components are connected through the messaging framework, publishing and subscribing to commands, events and telemetry, cooperating in a decoupled way to produce the Scheduler behavior. This design allows the deployment of the Scheduler in many ways. The components can be implemented in different computers, because the architecture remains the same regardless of the location of the actors, due to the communications bus implementation.

For example, each science program can be deployed in a different machine, real or virtual, provided it is connected in the messaging framework, subscribed to the corresponding telemetry, and publishing the proposed targets. New science programs could be incorporated without disrupting the live operations.

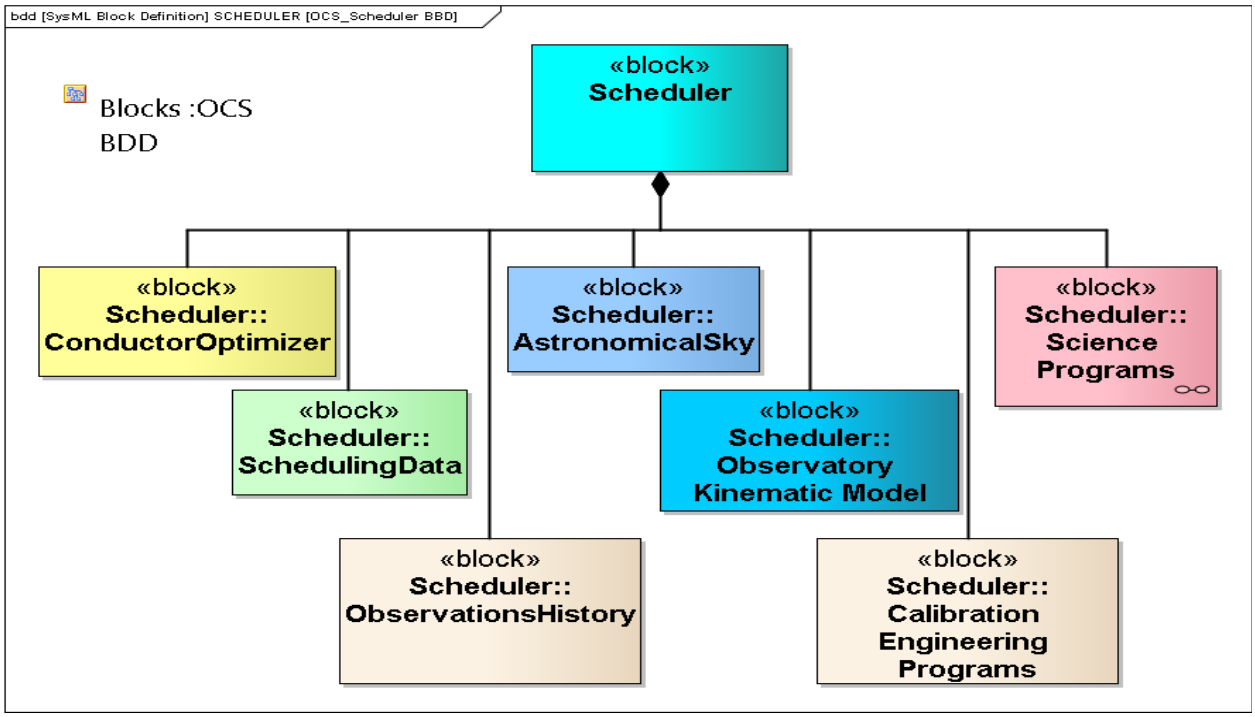

Figure 6. Block definition diagram for Scheduler.

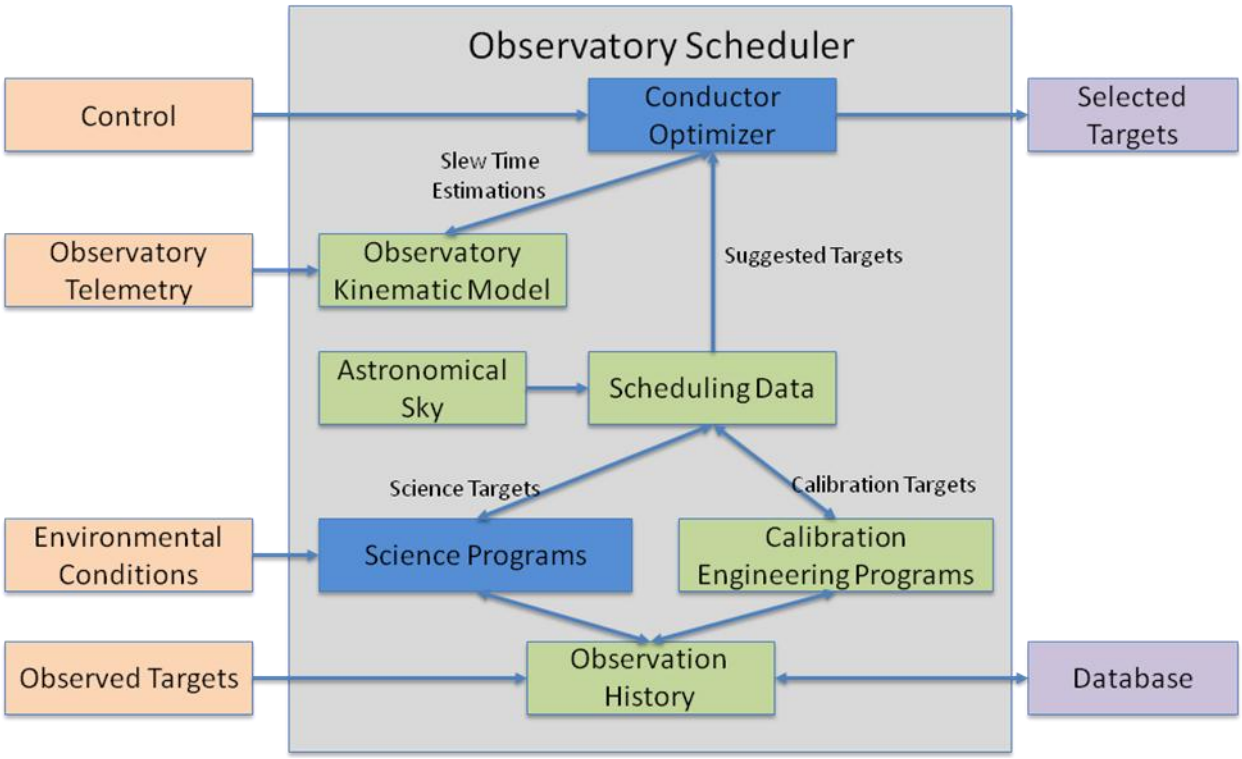

Figure 7. Basic internal data flows for the scheduler.

Science Programs

Multiple instances of this component can be included in the scheduler. The function of this block is to propose targets with the corresponding rank value. The particular sky region, ranking algorithm and parameters are configured in one configuration file per science program. The needed input data is the telemetry from the observatory and the environment, the previous visits from the Observation History, and the generated data from the models concentrated in Scheduling Data.

Scheduling Data

This is the component in charge of concentrating the observing parameters for the targets. The science programs publish their targets and information from the Astronomical Sky model and Observatory Kinematic model are added to the target matrixes. Look-ahead information concerning these parameters are pre-calculated for the configured time-window.

Astronomical Sky

Model for the astronomical sky, including coordinates for the sun, the moon, bright planets and sky brightness.

Observatory Kinematic Model

Model for the mount, rotator, dome, optics and camera, with enough detail to estimate the slew time to any target in advance for cost evaluation. This model has to match the performance of the real observatory, obtained from the telemetry.

Observation History

Local record for the past visits, including the parameters and conditions present at the time of the observations. This database is needed for ranking the future targets in the science programs, and to allow a warm start for any new science program after the survey has started.

Conductor Optimizer

This is the survey driver that collects all of the ranked targets from the science programs and applies the Observatory Kinematic Model to calculate the cost of each alternative. Given the value and the cost, and the available look-ahead information in Scheduling Data, an optimization algorithm is utilized to decide the schedule and publish the selected next target. Now, an important input is the visit that the OCS actually performed, to account for what in reality happened and adjust accordingly.

Calibration Engineering Programs

For other operational modes, such as engineering and calibration, components of the same type as the science programs support these observations with the same interfaces, but with different behavior and more operator intervention.

BEHAVIOR

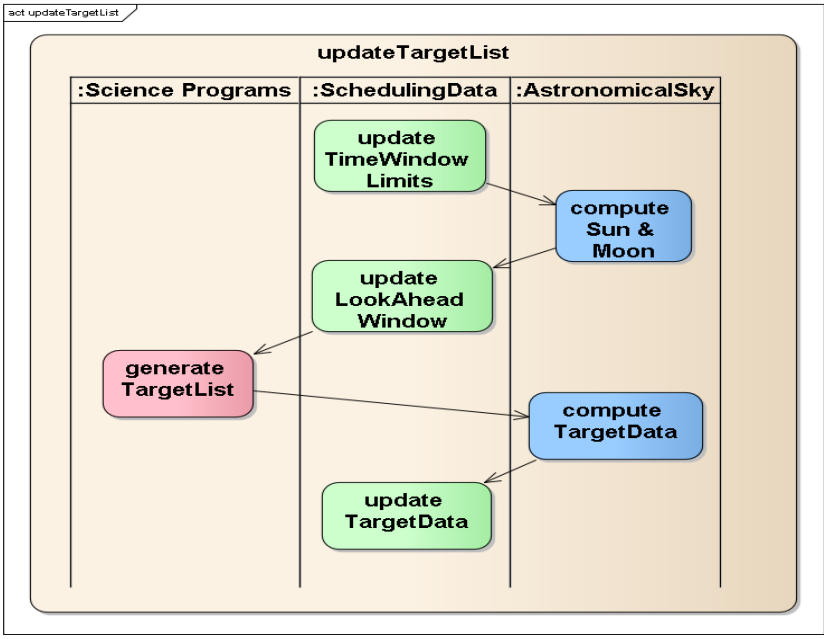

Figure 8. Activity diagram for the Scheduler, updating the list of targets at the beginning of the night.

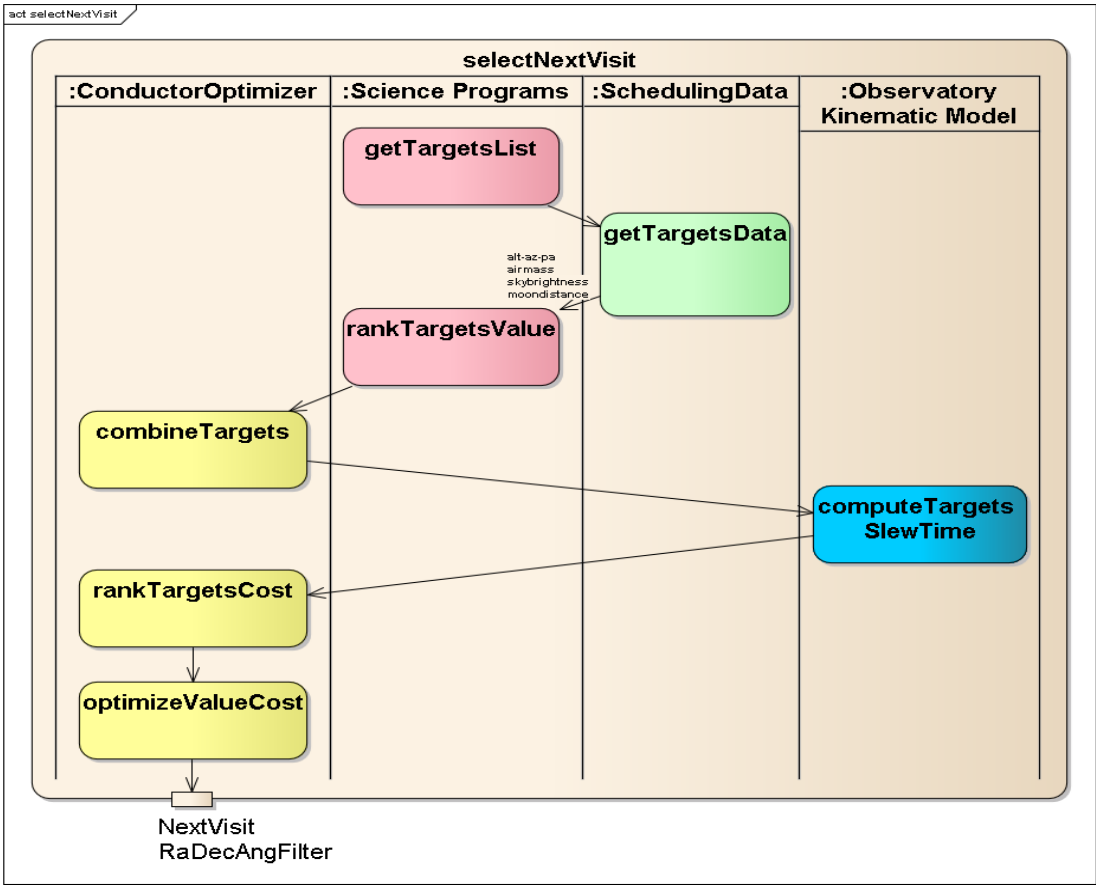

Figure 9. Activity diagram for the Scheduler, selecting the next visit at each single visit cycle in automated survey.

6. SUMMARY

The Scheduler design is well integrated with the overall OCS architecture, and complies with the requirements to fulfill the science goals of LSST. The partition of the functionality makes for a flexible implementation of the components, allowing the software to evolve and adapt to new requirements as they arise. The possibility to modify or add new science programs, include different scheduling algorithms, to configure the parameters for different strategies, give the needed flexibility. These characteristics have been exercised already in the scheduler prototype which is part of the LSST operations simulator.

The architecture and design allow for many options in the implementation and deployment. The scheduler is not tied to a language or platform. The communications middleware allows also to include any available telemetry as part of the scheduling evaluation process, provided a proper interface definition.

The selection of simple scheduling algorithms, is another advantage in the design. The simple concept of evaluating each target possibility with its value and cost, makes that the interaction of the competing science programs with their thousands of candidate targets at any given moment to produce an emergent behavior capable of solving the complex scheduling problem of the LSST mission. Many variants of the problem have been extensively tested in last years in the Operations Simulator.

7. REFERENCES

[1] Francisco Delgado, et al. "The LSST Operations Simulator". This proceedings.
[2] German Schumacher, et al. “The Large Synoptic Survey Telescope OCS and TCS models”. Proc SPIE vol 7738 (2010).
[3] Francisco Delgado, et al. “LSST Operation Simulator Implementation”. Proc SPIE vol 6270 (2006).
[4] German Schumacher, et al. “LSST Control System”. Proc SPIE vol 6274 (2006).