Agile development approach for the observatory control software of the DAG 4m telescope

SPIE. 9913 - 147

B. Bülent Güçsav*[a], Deniz Çoker [a], Cahit Yeşilyaprak [ab], Onur Keskin [c], Lorenzo Zago [d], Sinan K. Yerli [e]

[a]Atatürk Univ., Astrophysics Research and Application Center (ATASAM), Erzurum, Turkey; [b]Atatürk Univ., Sci. Faculty, Dept. of Astronomy and Astrophysics, Erzurum, Turkey; [c]FMV Işık Univ., Engineering Faculty, Dept. of Mechanical Eng., Istanbul, Turkey; [d]Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud, (HEIG-VD), Yverdon-les-Bains/Switzerland; [e]Orta Doğu Teknik Üniversitesi (METU), Physics Department, Ankara, Turkey.

OBSERVATORY CONTROL SYSTEM (OCS) SOFTWARE

The OCS software provides the enterprise level control of the DAG 4 mt class telescope and Focal Plane Instruments (FPI) and the observatory support systems. Following the requirements given in the Software Requirements Specifications[1] for Observatory Control System document draft and a series of meetings with DAG project team, three distinctive design constraints have been set initially.

DESIGN CONSTRAINTS

At first glance, the major constraints that should apply to a preliminary design is as follows :

- **Programmable control and monitoring interface:** The extent of the term *programming* describes an interface that makes any user to run automated observations from one or more Command Line Interfaces (CLI). Additionally, a batch-command execution environment will be available for sequenced jobs.
- **Concurrent users and simultaneous subsystems use:** The OCS software should support multiple user logins. Additional concurrency for multiple device/instrument access is also evident.
- **Configuration Management:** Run time configuration management for operational parameters as well as the behavior of the running software (i.e. it shall not be required to deploy the software on each configuration) should be present.

THE LIST OF EXISTING AND CANDIDATE SUBSYSTEMS *a subsystem is an entity accessible via OPC-UA connections

Service oriented approach as an initial choice of architecture, namely the industry standard OPC-UA[2], has been chosen for the low-level system access (Figure 1).

- Telescope Control System (TCS), active Optics Control System (aOCS), Enclosure Control System (ECS), Instrument Control System (ICS): Adaptive Optics Control System (AdOCS) & 6 FPI (at most) that can physically be arranged on two Nasmyth platforms . At this early stage, it seems that it would be beneficial to separate the ICS as a subsystem. The responsibility domain of this subsystem is to provide connectivity between the OCS and the various FPI and the necessary services for the OCS software. Science data header generation, depending on the particular FPI, will also be handled by the ICS. The science data header shall include: Instrument specific information, SEMS data for the particular observation or night, provided time information, primary subsystem information, session info. Wave Front Sensor (WFS): It requires to be a separate subsystem as it is going to be responsible for the high order control of the system as a whole. Site and Environment Monitoring System (SEMS): *Atmospheric Quality Measurements:* Various measurements will provide atmospheric quality data on a regular basis. *Meteorological:* Weather information data coming from various sources such as conventional meteorological devices and satellites. Collected weather data is going to be processed and will be sent to the OCS software as weather forecast information. *Seismic:* A system that collects seismic activity data from the stations around the DAG site. *Observation Site Facilities:* Infrastructures (power, communication, etc), surveillance and other support facilities.

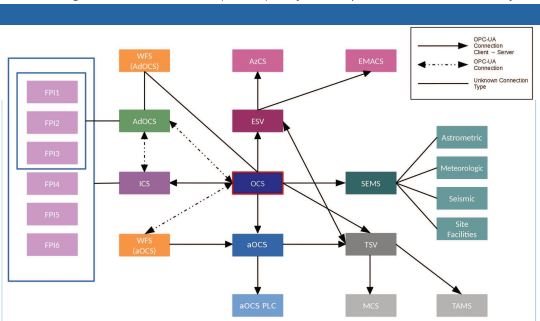

Figure 1. OPC-UA Client-Server layout of the subsystems. Directions of the arrows show the client-server relationships. Not yet known connections are shown without arrowheads.

USERS AND SESSIONS

One of the key concepts of the initial architecture is a *session*. A session is created on user login from a GUI or a CLI.

User Types	Control Primary Subsystem	Control Non-Primary ICS	Priority	Multiple User
Watchcat	✓	✓	NA	ONLY ONE
Operator	✓	✕	HIGH	ONLY ONE
Non-Operator	✕	✓	LOW	YES
Viewer	✕	✕	NA	YES
Engineer	✓	✓	HIGHEST	ONLY ONE

Table 1. Types of Users, their monitoring and control capabilities. This table shows all the users being online concurrently.

These are OCS internal users (Table 1). They can either directly be mapped to real world actors (i.e. human beings) or to some sort of robots that make the observations, calibrations, etc. on behalf of a real world actor. These mappings might or might not be in one to one correspondence with the actors specified in the Operations Concept Document (OCD).

OCS INTERNALS

The main idea is -in some sense- to provide a versatile intermediate service layer (or a service network) to the higher level systems such as Scheduler, post-processing scripts, batch command execution environment, etc. It is foreseen that the OCS software would mainly be developed on an application server. All the core functionality is executed by the components that collectively form a *core services framework* of the OCS. Each component family in the application server domain will implement an OPC-UA client per se to perform the necessary monitoring directly.

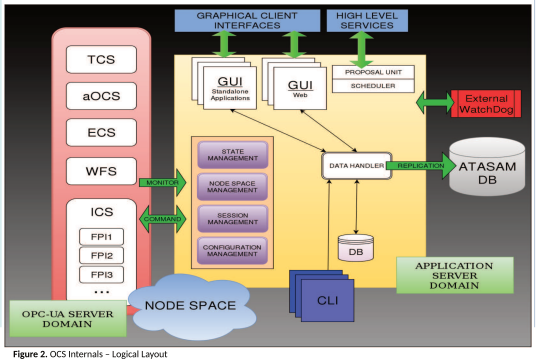

Figure 2. OCS Internals – Logical Layout

Command Line Interface (CLI): Instead of implementing a brand new command line interface for our own purposes, one viable option is to use a widely accepted and powerful tool already available. The inherent nature of the requirement has motivated us to think an interpreted programming language.

- *Direct* commanding of the subsystems
- Provide a programmable interface
- Users login through CLI → session
- Simultaneous CLI → session → multiple sessions

Graphical User Interface (GUI): Graphical accessibility(eg. Web etc.)

Interfaces to the higher level systems: Proposal Unit, Scheduler, etc.

Data Handler: A class of components for data storage and access

- A centralized logging system for both subsystems and OCS internals.
- Storage and accessibility of the ICS produced science data.
- Support for user configurable data formats for science data.
- Various forms of access to the stored data through CLI and GUI.
- An environment for pre-processing and post-processing of science data.
- Safe replication of all of the data to the tier 2 (ATASAM) storage.

Resource Management in the OCS: The union (logical aggregate) of all subsystems' OPC-UA address space nodes is called *node space*. It defines all the control and monitor domain for low-level systems of the OCS software. The grouping of services (and the actions taken place) include, session management (handles the requests from the GUI and the CLI, makes authentications and authorizes the user of a session by interacting with the proposal unit), node management (partitions the node space on request and assigns the demanded nodes to a session), configuration management (apply configurations that might apply to the current state and watch configuration update requests for a session or subsystem), and state management (coordinates the OCS and the subsystem states, constantly checks the in house components and communicates with the external watchdog, executes the routine procedures (Table 2) defined below.) (Figure 2)

Start-up	Shutdown	Safety/Emergency
• internal communication channels are active • all internal components are alive • the node space is accessible.	• All subsystems are operational • Bring all subsystems to "safe" state • Close all OPC-UA connections • Safely shutdown the OCS	• Not yet defined.

Table 2. Types of Users, their monitoring and control capabilities. This table shows all the users being online concurrently.

BEGINNING OF THE NEXT PHASE

- Solid understanding of the implementation details of the hypothetical CLI is crucial where the rapid prototyping plays an important role. To be more specific, the details of CLI integration with the core services must be clear.
- Although the interactions between the low-level systems is handled by OPC-UA, the heterogeneity in the implementation styles of the address spaces of different subsystems, push us to define an abstract common interface that mitigates the burden of coding difficulties. This abstraction layer might also give a better opportunity to implement any OPC-UA address space of a future instrument/system.
- Since not all of the FPI are known yet, the expected data to be produced is uncertain. Yet the technology to index data and provide an effective I/o speed for the DHS must be considered.

REFERENCES

1. Gucsav, B.B., et al. DAG int. doc. (2016)
2. OPC Foundation, https://opcfoundation.org/developer-tools/specifications-unified-architecture
3. Coker, D., et al. DAG int. doc. (2016)

Authors would like to thank, Republic of Turkey, Ministry of Development; Atatürk University, Erzurum/Turkey (Project No: 2011K120230); Atatürk University, Erzurum/Turkey, Astrophysics Research and Application Center (ATASAM), Erzurum/Turkey; Orta Doğu Teknik Üniversitesi, Ankara/Turkey; FMV Işık University, Istanbul/Turkey; FMV Işık University, Center of Optomechatronics Application and Research (OPAM), Istanbul/Turkey; Haute Ecole d'IngÉnierie et de Gestion du Canton de Vaud (HEIG-VD), Yverdon-les-Bains/Switzerland, for their support throughout the DAG project.