Observatory Science Operations Tool Development for the SKA within a Scaled Agile Framework

Pamela D. Klaassen[a], Stewart J. Williams[a], Mark Nicol[a], Valentina Alberti[b], Alan Bridger[a,c], Antonio Chrysostomou[c], Snehal Valame[d], Elizabeth S. Bartlett[a], Matteo Canzari[e], Abhijeet Deolalikar[d], John Lightfoot[a], Andrew McDermott[f], Viivi Pursiainen[a], Hélder Ribeiro[g], and Jose Sabater[a]

[a]UK Astronomy Technology Centre, Royal Observatory Edinburgh, Blackford Hill, Edinburgh EH9 3HJ, UK
[b]INAF - OATs, Trieste, Italy
[c] SKA Organisation, Jodrell Bank Observatory, Cheshire, SK11 9FT, UK
[d]Persistent Systems Ltd, Pune, India
[e]INAF, Osservatorio Astronomico d'Abruzzo, Teramo, Italy
[f]Rutherford Appleton Laboratory, Harwell Campus, Didcot, OX11 0QX, UK
[g]FCUP - Centro de Investigação em Ciências Geo-Espaciais (CICGE), Portugal

ABSTRACT

The SKA requires a comprehensive suite of applications to be prototyped in the current 'bridging' phase ahead of the formal start of construction, leading to full development in the subsequent construction phase. The Scaled Agile Framework (SAFe®) has become an industry standard process for managing the development of large software systems, defining a set of processes to manage and coordinate the activity of multiple agile software development teams. SAFe has been adopted by the SKA, and is being used to coordinate a large number of globally distributed agile development teams; including the team developing prototypes of the Observatory Science Operation (OSO) applications.

Much of the team who developed the OSO design for Critical Design Review (CDR) are now involved in the agile development of the OSO tools, and with the shift to SAFe development have a unique view on how to develop within SAFe from a plan that was developed anticipating a traditional waterfall software development process. Here we present an overview of how evolutionary prototypes of these tools (from proposal handling and assessment, through to observation design, planning, scheduling and execution) are being developed for the SKA within SAFe.

Keywords: SKA, Observatory Control Systems, Scaled Agile Framework, Software Development

1. INTRODUCTION

For world class observatories like the Square Kilometer Array (SKA), automating as many processes and procedures as possible ensures the smoothest and most efficient use of telescope time, and therefore maximising the scientific return on investment. For the SKA, the science operations tools are being developed with this framework in mind; to minimise the manual interactions required to run the observatory and thus, the possibilities of inserting errors into the process. Here, we define the observatory science operations (OSO) tools as those facing the users (scientists and engineers) that take an observational project from inception to data processing.

In the current bridging (pre-construction) phase of SKA development, the software teams are de-risking issues found within the critical design reviews of the various systems, and here we present the current workings

Further author information: (Send correspondence to A.A.A.)
A.A.A.: E-mail: pamela.klaassen@stfc.ac.uk, Telephone: +44 0131 6688 218

Observatory Operations: Strategies, Processes, and Systems VIII, edited by David S. Adler, Robert L. Seaman, Chris R. Benn, Proc. of SPIE Vol. 11449, 114490Z · © 2020 SPIE
CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2562109

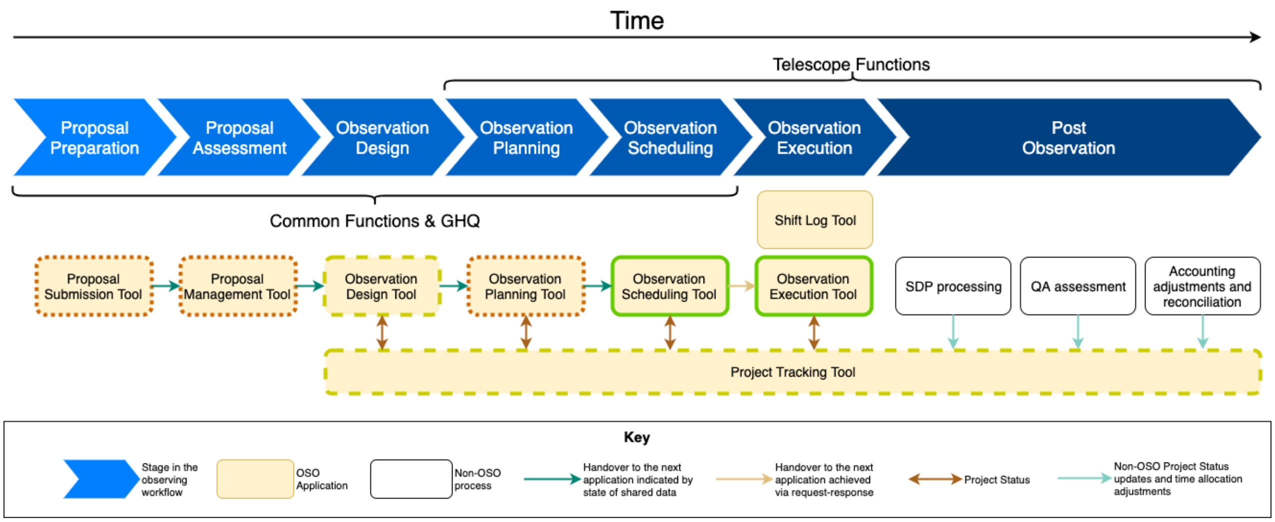

Figure 1. Overview of OSO tools and the workflow items they represent. Those tools in orange (dashed) boxes are hosted at the SKA headquarters, while those with green (solid) boxes are deployed from the MID and LOW sites. Those in yellow (dotted) are replicated between all sites.

for the OSO tools being prototyped during the bridging phase. Along with the other software systems being designed for the SKA, the OSO tools are expected to be built within a Scaled Agile Framework (SAFe®).

Below, we introduce the architecture for the OSO tools, and describe how data flows through the tools. We then describe how these tools are being developed within SAFe, and what that means for incremental development of a *set* of tools. We follow this with a brief note on the lessons learned so far by the OSO tools team, and the outlook for tool development in the near future.

2. THE OBSERVATORY SCIENCE OPERATIONS TOOLS

Within the SKA ecosystem, a scheduling block (SB) in the quantum of schedulable, self contained observing, while a processing block is the equivalent for the processing of observations. Instances of Schedluing blocks (SBIs) are executed on the system once the observations have been scheduled and late binding information included in the instance. Multiple SBIs can be generated from a single SB with differing sets of, for instance, calibrators or order of executions depending on source availability.

As an observing project moves through its lifecycle from initial proposal to all observations being complete and data processed, the scientists, engineers and technicians involved interact with the Observatory Science Operations (OSO) tools developed to bring it through the system. Figure 1 shows the workflow through a project with the corresponding OSO tool that manages it. Below, we outline the tasks performed by each of the tools as a project moves through this system.

2.1 Proposal Preparation and Management

The first tools an proposer will use when interfacing with the SKA system are those responsible for proposal preparation, submission and management. Under the umbrella of a generalised Proposal Handling Tool (PHT) for the 'Phase 1' portion of the project lifecycle, there are the Proposal Submission Tool (PST), and the Proposal Management Tool (PMT).

The PST will consist of the user interface and submission system. This tool will allow users to input their proposed observation plan, type of observation, time on sky, computing resource requirements, and their science/technical justification for review. The submission system will be used for uploading the proposal to the SKA archives. The PMT will be the tool used for proposal review.

In both parts of the PHT, users will have access to distinct parts of the tools depending on their roles within the project and the observatory (e.g. proposal PI, proposal reviewer or SKA staff). The options within the PHT will change from cycle to cycle as telescope capabilities change and as will be advertised in the Call for Proposals.

One a proposal has been approved, it becomes a project, and the 'Phase 2' portion of the project lifecycle begins: observation design. This is done using the observatory tools, described below.

2.2 The Observatory Tools

The next set of tools are those involved in observation design, planning, scheduling and execution (the ODT, OPT, OST and OET, respectively). These tools convert the science requirements detailed in the observation design to telescope configurations that are executable, schedulable, and can be tracked.

The ODT creates scheduling blocks: self contained sets of observations that contain all of the information required to manage observing and data processing resources, pointing information and observations of sufficient calibrators (whether in field or requiring separate observations) to do the data processing. The components of a scheduling block are used to create executable lists of commands to be executed at the telescopes once scheduled.

The OPT derives an executable observing plan over a set timescale (e.g. a full observing cycle, or for the next month) consisting of a prioritised list of scheduling and processing blocks that can fit within the timescale. It must include models for the expected weather and resource availability over the planning timescale, including any maintenance or other expected downtime during the interval. It takes into account resource management to determine which projects can be carried out simultaneously, either commensally or in different sub-arrays, and whether the Science Data Processor (SDP) has enough capacity to process the data. What it produces is an executable list of prioritised scheduling blocks which can be scheduled for execution.

The OST creates a near real-time observing schedule from the prioritised SB list created by the OPT. It creates executable versions of the scheduling blocks with late binding information such as (potential) observation start time, and available resources and calibrators. The exact durations of these blocks can then be determined and placed into a telescope schedule. This schedule is then optimised against observing pressure, priority and resource availability on the short (days) and medium (weeks) term. The optimised schedule can then be sent to the telescopes for execution

The OET translates the scheduling blocks (SBs) into scheduling block instances (SBIs) which can be run on the telescope. If required by the project, individual SBs can be run multiple times as separate SBIs with the timing and target information relevant for a specific execution injected into the SBI. The OET then translates the commands in the SBI from the scientific domain to the telescope domain for execution by the system. It sends the commands to the telescope manager[1] which then relays the appropriate commands to the relevant telescope sub-systems. It also monitors this process and reports the status back to the Project Tracking tool via the OSO Data Archive (ODA)

2.3 The Project Tracking Tool and Archive

To have oversight of the observatory tools, there needs to be a project tracking tool (PTT) which determines and stores the state of any and all projects planned, executed or completed by the observatory. This tool tracks how many times an observation has been executed, how much observing time is remaining on a project, what resources have been consumed, the status of SDP processing for the project, any QA metrics and project metadata such as PI and what kind of project it is (Key science, principle investigator designed, targets of oportunity, etc.).

The data archive stores all of the artefacts related to projects, observing, and data processing. High availability is a key requirement for the ODA to ensure that the databases across all SKA sites (the global headquarters, and both the MID and LOW telescopes) remain in sync.

2.4 Initial Decisions in Deployment and Interfaces

A number of key decisions were made with respect to the software architecture in the CDR, including using a shared data architecture for the OSO tools, and to have individual tools for each workflow stage (See Figure 1). This is common practice for most modern observatories and allows for flexibility and future modifiability within each tool.

The proposal handling, observation design and planning will be managed at the SKA Global Headquarters (GHQ), and other tasks such as observation scheduling and execution will be managed at the telescope sites (See Figure **??**). This requirement mandates excellent availability and replication of information between the sites, and that the observation design tool and project trackers, which act as the interface between observation planning and scheduling, will need to be highly available and up to date with the data archive. This allows for observation design, planning and scheduling can all be done at each site if and when required.

As per the CDR documentation, the OSO tools are required to conform to a number of quality attributes, including performance, availability, modifiability, usability, interoperability, robustness and security. The importance of these attributes varies with the tool and its intended use. For instance, the availability attribute of the ODA is much more stringent a constraint than the availability of the OPT.

The usability of the proposer/user facing tools (i.e. proposal preparation and submission, project design and tracking) is of high value, and these tools will be created using web-based technology to ensure platform independent performance.

3. SCALED AGILE FRAMEWORK

The adoption of SAFe for SKA software development itself is described in more detail in Bartolini et al. (these proceedings). Here it is important to note that in the current pre-construction phase of SKA software development, there is a single team developing the OSO tools, and those resources will be increased to two teams after the transition to construction.

The overall prioritisation of prototype development activities is decided by the SKA office for each quarterly Programme Increment (PI). The OSO development team is then responsible for the technical breakdown of the work into small enough tasks to be completed by individual developers within each two week long sprint. Within the planning meeting the team, in collaboration with the SKAO representatives defines the scope of the work to be undertaken and sets expected delivery dates for the individual activities. The distributed nature of the OSO team means that these planning events are the only chance the team has to meet face-to-face.

4. BUILDING THE OSO TOOLS WITHIN SAFe

The primary driver of pre-construction software development is to de-risk the emerging system against concerns raised during CDR, and building up the construction-ready SAFe teams. For OSO tool development, this means building evolutionary prototypes of the tools so that the Prototype System Integrations (PSIs) for the Low and Mid Telescopes can be exercised. It also means forming effective agile development teams working with SAFe that are capable of taking SKA forward through construction. Because the OET is so central to the OSO framework, and the overall execution of observations in the PSIs, it was the first of the OSO tools to be prototyped.

Development began with a rudimentary OET which is able to pass commands to the emerging Telescope Monitoring and Control (TMC) systems.[1] As shown in Figure 3, use of the OET itself (bottom left corner) requires being able to load in minimal scheduling blocks (SBs). To efficiently and consistently create valid SBs, we have also begun development of the ODT and prototyping an ODA / file store for those scheduling blocks.

Within the SAFe framework, prioritisation of ODT and ODA component implementations flowed from the requirements placed on the OET as the downstream system prototypes gain functionality (i.e. what commands it needs to send to TMC).

The requirement to start prototyping the ODT comes from being able to create reliable, reproducible, and complete SBs for ingest by the OET. With a rudimentary ODT in place SBs can be generated and stored

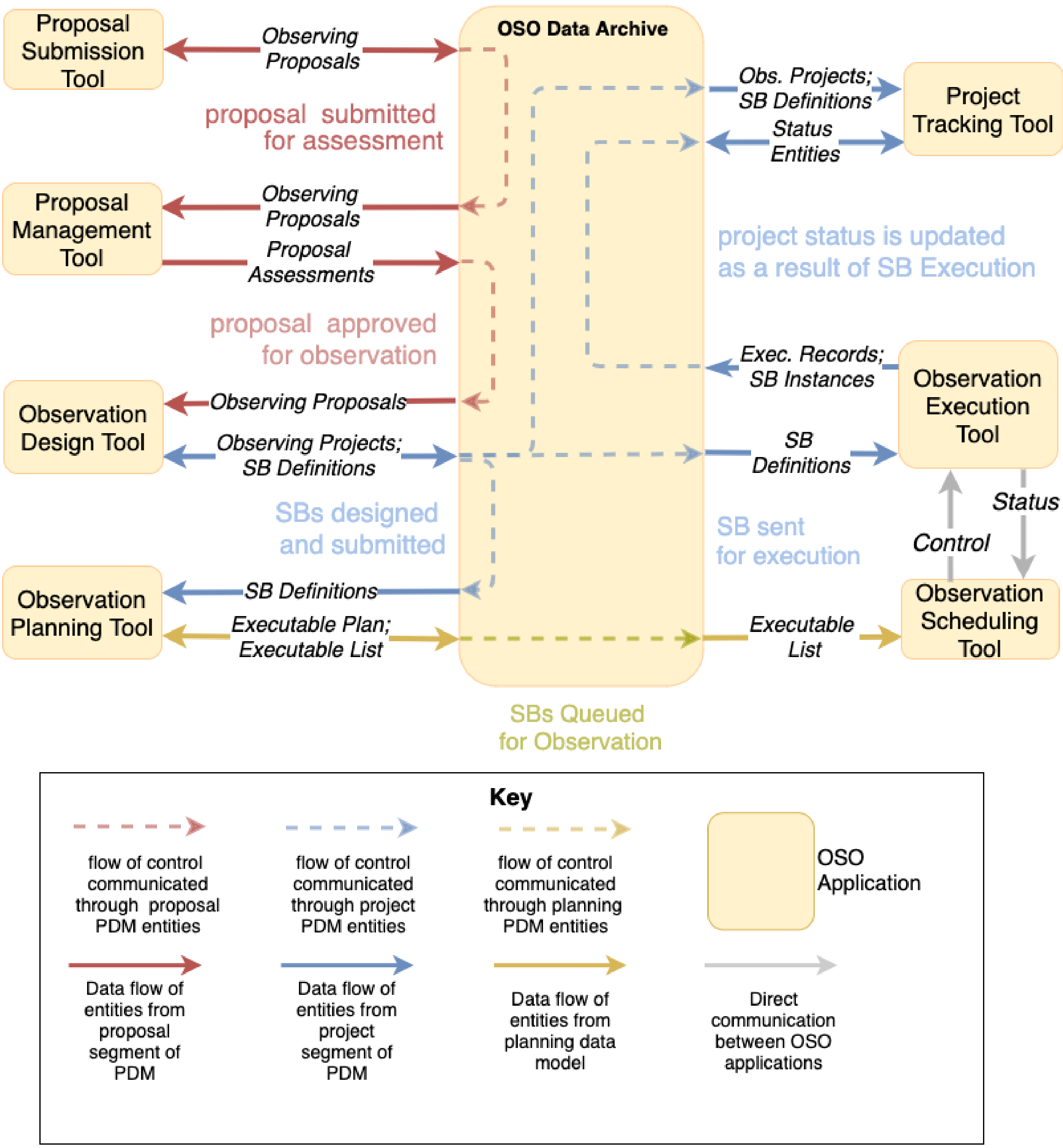

Figure 2. Overview of OSO tools and their relationship with the Data archive.

for the OET to ingest as will be done during construction and normal operations. SB generation, storage and retrieval becomes more important as we move towards Prototype System Integrations (PSI) for both the Low and Mid telescopes. With these pieces in place, the prototype system can be rapidly, consistently and repeatedly exercised.

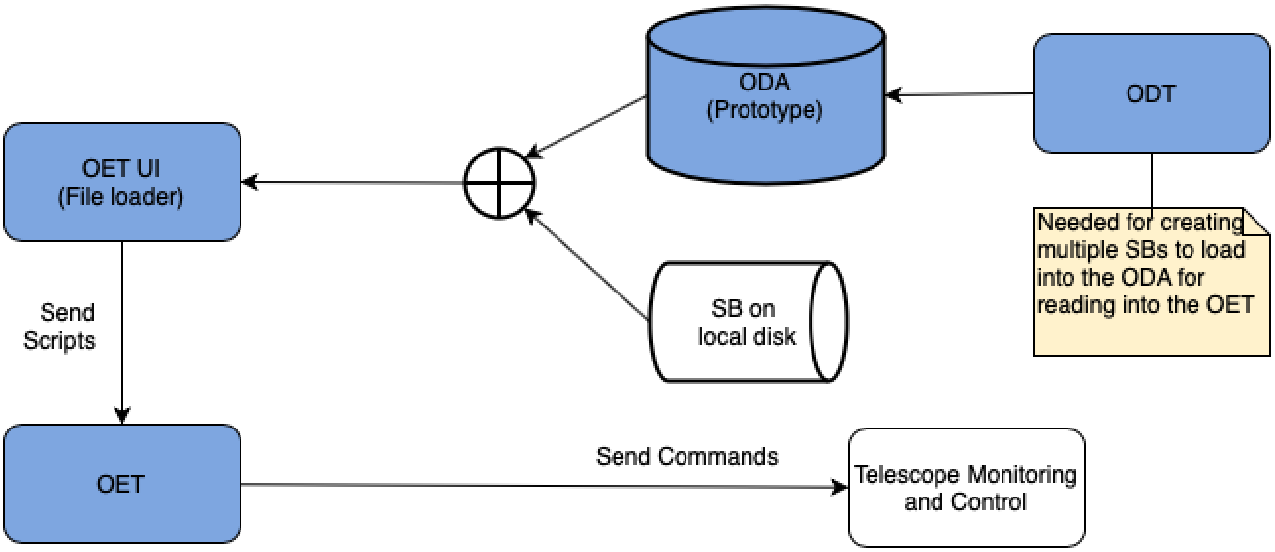

Figure 3. Example of the minimum viable products for the OSO tools being prototyped during the pre-construction phase. The Observation Execution Tool (OET) has been prioritised for initial prototyping, and to facilitate exercising its interfaces, rudimentary scheduling blocks (SBs) are required. To avoid repetition and potential user errors, the basics of an Observation Design Tool (ODT) are also being prototyped to create those SBs.

5. LESSONS LEARNED

While the team is still in its early stages of adopting SAFe (< 10 programme increments), there are already a few lessons being taken on board with respect to the setup of the project, the distributed nature of the development, and how the various teams in the Agile Release Train interact at development boundaries.

Having a solid foundation from which to build a development roadmap has been of tremendous value to keeping the prioritisation (and thus development) planning on track. Keeeping an eye on the intended functionality of the different tools has allowed the team to anticipate what functionality, and from which tools, will be required next to allow the system prototypes to show their progression towards full telescope systems. Figure 3 shows this in action. The PSIs require a minimal observation execution tool, and to populate it with scheduling blocks requires: 1) an observation design tool capable of creating validated minimal viable scheduling blocks, 2) a minimal observational data archive to store the created SBs, and 3) a definition of the minimum viable scheduling blocks to be built. Because each of those tools and products were envisioned in the documentation delivered for CDR, creating the workflow in Figure 3 required minimal decision making at the prioritisation meetings.

Similarly, having well defined APIs to the products being developed by other teams means that the teams only need to discuss potential changes to APIs at the planning meetings, rather than wholescale interfaces - which would require higher level approval and checks for consistency across the entire system.

Many of the decision makers involved in the OSO tool development were involved in creating the documentation submitted to the CDR process, and as such, are able to properly prioritise foundational work on the tools: even though essential groundwork might not have the highest short-term business value, how much risk is being mitigated by enabling work is being properly estimated in the prioritisation exercises.

A key lesson that has emmerged from these experiences is that while SKA prioritisation for development (neccessarily) focuses on incremental value being added to the overall system, the team needs to balance that with any groundwork that could minimise risk and complexity in the long term.

6. DISCUSSION AND CONCLUSIONS

As we move towards formal SKA construction, a lot of the risk in the OSO tools is being retired as the system as a whole is beginning to take shape. Because the responsibilities of, and boundaries between, the OSO tools were well defined from the outset, there is a clear and prioritised roadmap for development that is able to accommodate the changing landscape of SKA software development as the system evolves.

REFERENCES

[1] Bridger, A., Gupta, Y., Chaudhuri, S. R., Di Carlo, M., Le Roux, G., Natarajan, S., Smareglia, R., Patil, M., Barbosa, D., van den Heever, L., Dolci, M., Alberti, V., Brederode, R., Barraca, J. P., Bartashevich, D., Bergano, M., Brajnik, G., Canzari, M., Dange, A., Guzman, J. C., Jerse, G., Khanvilkar, A., Klaassen, P., Knapic, C., Kodikar, J., Kumthekar, V., Maia, D., Mohile, V., Morgado, B., Nakave, S., Nicol, M., O'Brien, A., Ramanujam, N. M., Ranpura, J., Reed, S., Sathe, V., Silva, N., Swart, P., Tinarelli, F., Trivedi, V., Babani, L., Valame, S., Vrcic, S., Williams, S., and Wadadekar, Y., "SKA telescope manager: a status update," in [*Software and Cyberinfrastructure for Astronomy V*], *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **10707**, 1070703 (July 2018).