Leverage collaborative design techniques to develop a unified interface for the SKA Central Signal Processor

Valentina Alberti[a], Gianluca Marotta[b], Matteo Canzari[c], Grant Hampson[d], Andrew Bolin[d], and Giorgio Brajnik[e]

[a]INAF-OATs, Via G.B. Tiepolo, 11, 34143,Trieste, Italy
[b]INAF-OA Arcetri, Largo Enrico Fermi, 5, 50125 Firenze, Italy
[c]INAF - OAA, Loc. Collurania, Via Mentore Maggini, snc, 64100 Teramo, Italy
[d]CSIRO, Space & Astronomy, 26 Pembroke Road, Marsfield, NSW, 2122, Australia
[e]Interaction Design Solutions and Dip. di Scienze Matematiche, Fisiche e Informatiche, Univ. of Udine, Udine, Italy

ABSTRACT

The Square Kilometre Array (SKA) Central Signal Processor (CSP) is a real-time backend system that processes incoming astronomical signals to produce visibilities and detects and profiles pulsars. The CSP is composed of the Local Monitoring and Control (LMC), the Correlator and Beam-Former (CBF), and the Pulsar Search and Timing (PSS, PST) engines. Each subsystem is developed by a different team in the SKA control software domain following the Scaled Agile Framework (SAFe) to guarantee coherence in the development. The definition of an engineering User Interface (UI) for the CSP is challenging due to the variety of skills that are required to identify the most relevant design concepts and potential roadblocks to an effective representation and the fact that several teams are involved. For this reason, we chose to leverage a collaborative design approach that can easily fit SKA's biweekly sprint cadence while involving experts from different fields in a "think outside the box" process. Sketches and wireframes undergo multiple refinement sessions that lead to the realisation of an engineering dashboard representing the current state of CSP implementation. User testing sessions constitute the means by which the success of the proposed UI is measured. Additional positive effects are alignment across different teams on the current capabilities of the system and its future development, as well as a way for continuously adapting the UI to the system's evolution. In this paper, we describe the challenges we faced while coordinating the design across multiple teams, show how the process was implemented to fit the short agile iterations and overall SAFe framework and present the results of the work.

Keywords: Lean UX, User Experience, Usability, User Interfaces, Agile Software Development, Square Kilometre Array, Radio-telescopes

1. INTRODUCTION

As the first Array Assembly release (AA0.5)[1] of the SKA telescopes[*] is approaching, the interest in User Interface (UI) development has seen constant growth and multiple initiatives have been undertaken to tackle this large-scale problem. Human interaction is in fact expected at various levels, from submitting proposals to scheduling observation and telescope operations, to monitoring the system health and system control. This last area is the one that needs to be ready sooner to allow integration teams and testers to stress the potentiality of the delivered hardware (HW) and software(SW) components, to verify their adherence to the requirements, to explore critical paths and finally to validate the product. We are approaching the challenge of creating engineering UIs from

Further author information:
Valentina Alberti: E-mail: valentina.alberti@inaf.it

[*]The SKA Observatory comprises two telescopes that will operate at different frequency ranges. The Mid telescope, 197 dishes in the Karoo desert, South Africa, will observe the sky in the frequency range 350MHz - 15GHz, while the operational frequency range of the Low telescope in the Murchison Radio-astronomy Observatory, Western Australia will be 5 - 350MHz.

Software and Cyberinfrastructure for Astronomy VIII, edited by Jorge Ibsen, Gianluca Chiozzi, Proc. of SPIE Vol. 13101, 131014I · © 2024 SPIE
0277-786X · doi: 10.1117/12.3021688

the point of view of developers of the Central Signal Processor (CSP), the subsystem in charge of collecting the signal from the receptors, channelising it, synchronising it in real-time. The system is composed of three engines, the Correlator and Beam Former (CBF) which produces complex flux measurements in spatial frequency space (visibilities), Pulsar Search (PSS) which identifies pulsar candidates, and Pulsar Timing (PST) that precisely calculates the period of the identified pulsars as well as any irregularities. The Local Monitoring and Control (LMC) element of the CSP assures that the bidirectional communication between the three engines and the Telescope Manager happens coherently, exposing the CSP as a single entity[†]. Teams from different countries and belonging to different management streams (called Agile Release Train[2,3] in SAFe, the agile framework adopted by Square Kilometre Arry Observatory (SKAO)) develop each component. We describe the approach we followed to move a step forward from,[4] by increasing the scope of the dashboards to include the CBF of the low-frequency telescope and by aiming at a higher quality level.

2. CONTEXT

Defining a user interface based on a set of dashboards for the SKA-Low CSP presents the following challenges.

Complexity of the SKA-Low CSP system: The CBF is the core engine of the SKA-Low CSP. It provides the capability of processing interferometric data from 512 stations (each comprising 256 antennas) in real time to create visibilities and beams that enable further scientific processing. Eleven different processing modes allow for creating up to 500 pulsar search and 16 pulsar timing beams and 2 image modes. Depending on the observation requirements, stations can be split into substations and the final configuration can be grouped into 16 subarrays. The processing is carried out by specific HW components (Alveo cards) connected to P4 network switches. The full HW stack is quite complex. More details can be found in.[5] Our work focuses on the LMC and CBF engines and a configuration with 6 stations, 4 processors, and 4 subarrays (which is what will be deployed at AA0.5, the first major milestone in the telescopes' construction phase).

Evolutionary phase of the project: SKA project is now at the beginning of the construction of the telescopes, in a phase where the system will rapidly evolve from the current stage of verification and commissioning to a scientific instrument. The currently immature CSP engines will quickly progress towards more complete implementations. Dashboards should follow the same growth path, serving different users with different goals over time, starting with the early-stage operations and engineering qualification, which is our current focus.

Complexity of the social environment/process/management framework: The SKA project is now facing the first Array Assembly release, AA0.5. As the amount of required effort is huge, the project as a whole includes about 30 development teams grouped in different Agile Release Trains each with their own management structure. All the teams plan their work every three months and release chunks of working software every two weeks. The schedule is quite tight and priorities need to line up in a coordinated effort that leads to delivering a working integrated system for major verification events. In this context, raising the priority of developing UIs to the top of the list may not be straightforward. For this reason, it is crucial to define a sustainable method to advance and refine complex UIs.

Lack of a standard approach: The development of UIs is an emerging topic in SKA. Various teams and experts are tackling the problem and exploring solutions but coordination is currently lacking. Not being able to rely on a well-established approach we decided to contribute to the effort by proposing and validating one. Given the agile nature of the project management, we naturally looked at Lean UX[6] as a source of inspiration. Despite its concrete and iterative nature, following it without tuning it to our needs seemed unfeasible for the following reasons: competing project priorities which translated into limits to the capacity that each team can dedicate to this effort, unfortunate timezone difference for the involved teams (Italy and Australia), and lack of experience in designing complex UIs. We describe the process we followed and the obtained results in the following sections.

[†]This description holds for both Mid and Low telescopes

3. ADOPTED APPROACH

Given the complexity of the problem and the need to adopt an agile approach, we decided to rely on techniques derived from the Lean UX method.[6] Lean UX is recognised as the way to bring the historically monolithic and long-lasting design process to an iterative cycle of generation of ideas and verification steps. It offers a process to recursively explore design solutions developing the minimum amount of artifacts needed to move forward in understanding the product we are developing. It leverages the power of collaboration and cross-functional teams and enforces the benefits of making ideas visible as soon as possible to select the most promising or urgent ones and focus the development and validation process to verify the correctness of the underlying assumptions. The feedback received during the validation phase helps to refine the product.
The Lean UX process is thoroughly detailed in.[6] After studying it we came up with our selection and interpretation of the proposed steps and techniques that fitted our schedule and skills at best. In particular, we wanted to reduce to the minimum the preparatory work and start the collaborative design as soon as possible.
The first step was to find the right participants. We created a cross-functional team of 6 people including experts of the subsystems, of Taranta[7,8‡] and of Lean UX and UI development. The proposed process included the 4 phases described below.

Phase 1: Define and agree on the vision for the work. The first step of the process aimed at reaching a shared vision of what we wanted to achieve, with clear ideas of its scope, motivations, and outcomes. Aligning the team to a mission is a critical part of the process that enables moving towards the same direction while going through the subsequent phases. After the initial brainstorming, we converged on the following definitions.

Problem statement: engineering UIs are currently missing for the SKA CSP subsystem despite being very much needed. In this initial effort, we target the LMC and CBF, leaving the other 2 subsystems (PSS and PST) out of scope, and we mainly focus on expert users, generally identified as troubleshooters.

Users and Use Cases (UCs): users of engineering dashboards are subsystems' experts or integration teams trying to test the various components. They want to identify issues very rapidly and dig down exploring more details if needed. To do so they need to be aware of the overall system's health, of what operations it is executing on what HW components, and if the data flow is as expected. Additional UCs were identified but deemed less urgent. The UI should provide minimum guidance to the users who may be experts on a specific part of the system but less knowledgeable about others.

Project outcomes: this step answers the following question: "what positive changes do we expect to see in the overall SKA project after we develop effective UIs?" We agreed on the following answer:

- other teams and stakeholders use the provided UIs instead of building their own, ad hoc, short-lived solutions;
- system bugs are promptly found and flagged;
- a broader audience understands the connection between different CSP components.

User outcomes: if we develop effective UIs, what will the users of the UIs do differently? We expect that the users will move from extensively using the command line tools to a more graphical UI, that their confidence in the system's behaviour will increase as well as their understanding of the less familiar components and how the various pieces are connected. Moreover, code quality improvements are enabled by the easiness of identifying misbehaviour in the system, executing unexpected sequences of commands and checking the result.

Phase 2: Generate and select ideas. After agreeing on the vision we run a set of collaborative design sessions with the following schema: each participant chooses one UC they are interested in and draws a rough paper prototype that is subsequently shared with the group. Control software experts turn into designers and explain the mock-ups, propose visualization ideas, and constructively review the work of others. UX experts ask probing questions to flash out underlying assumptions about the behaviour of the system, the users or the UI.

‡Taranta is the tool identified to develop engineering UIs within SKAO

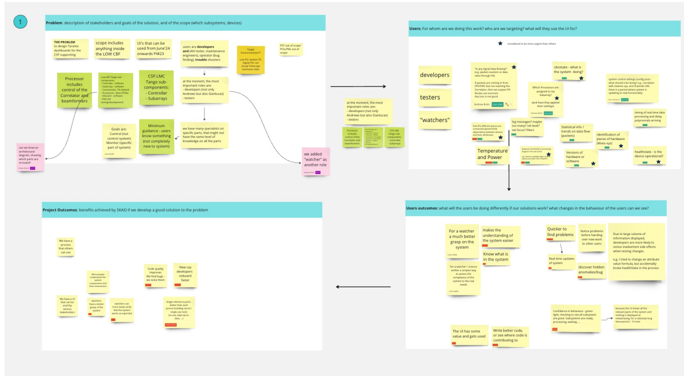

Figure 1. Output of Phase 1 as it appears on our virtual board

This session is relatively short and repeated twice to cover as many UCs as possible. The following step is to evolve the mock-ups to medium-fidelity ones and focus on the most relevant UCs to flash out more details, what control commands and attributes are involved, and what workflow would a user follow to get certain information.

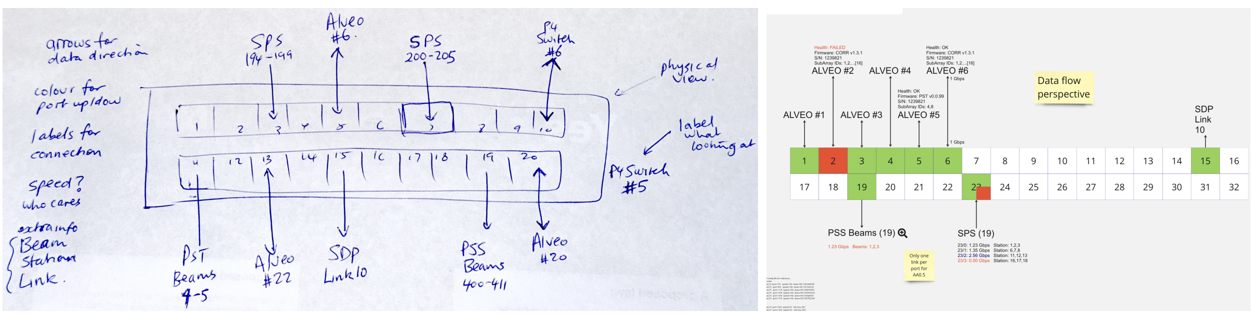

Figure 2. Examples of low and medium fidelity prototypes

Phase 3: UI implementation in Taranta. To create web-based dashboards in Taranta is a straightforward process. Taranta provides a set of widgets that can be dragged and dropped on the canvas area and configured to connect with Tango devices. To prepare for a user testing session, the style of the dashboards has to be fleshed out so that it is coherent across the set and limits the visual obstacles as much as possible. A UX person took up this task and tuned the design using as an input a colour schema suggested by other teams in SKA, the medium fidelity mock-ups and the dashboards already used by the LMC team.[4]
Phases 2 and 3 were important to identify improvements in the exposure of Tango attributes by the subsystems that would simplify the access to relevant information from a client (Taranta or other subsystems) perspective.

Phase 4: Internal validation and preparation for the usability investigations.[9] This final stage consists of validating the developed UIs first by the LMC and CBF control software developers who participated in this work during their normal development activities, and then by an external tester or troubleshooter. It is crucial to arrive at the user testing session (one of the usability investigation techniques that we plan to adopt) well-prepared. The team identified the following success criteria to be ready to run the session:

- A set of tested dashboards that show coherent style and behaviour and that support the envisaged UCs.
- Well-defined objectives for the session in terms of actions and decisions that the tester should be able to perform and success criteria (i.e. were the tasks completed successfully? Were errors experienced? When? Could the user recover? How long did the task take? Etc.)

- A properly configured environment in which the tasks can be performed.
- An experienced mediator to run the session.

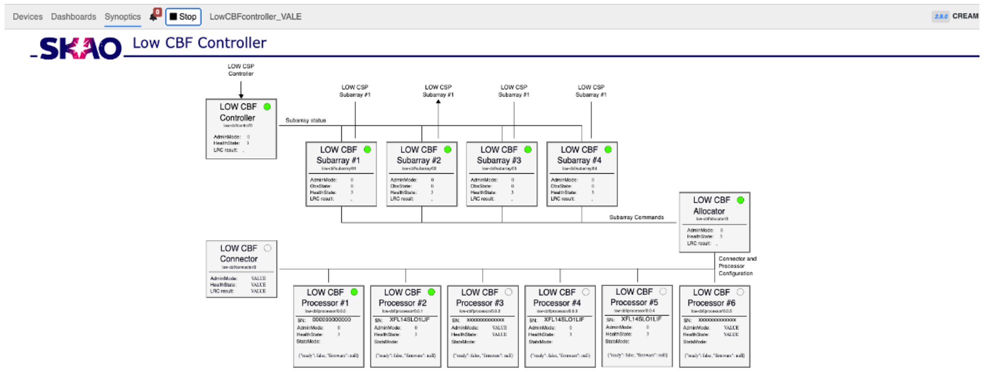

Figure 3. Examples of a dashboard created using Taranta synoptic, showing the CBF controller structure

4. CONCLUSIONS

Inspired by Lean UX, we approached the problem of defining a sustainable process to create and evolve UIs for the SKA CSP taking into account the boundary conditions posed by the project's priorities, the complexity of the subsystem and the available expertise and time. We developed a process that guided us through the development of a set of engineering dashboards to support troubleshooters of the LMC and CBF components. The set partially covers the identified high-priority use cases, has a coherent style and is currently under internal validation. Improvements have been identified in the following three areas:

- Process: increase the sessions' duration; prepare a short checklist of the needed dashboards and supported functionalities to ease the check against UCs; follow a more formal validation process; share a standard environment that can also be used for testing sessions.
- Code: refactor the code to expose additional data to increase the ability to debug the system through the UI (often such data will be useful to other components as well).
- Tool: developing Taranta dashboards revealed some usability and functional improvements to the currently available widgets as well as requests for additional functionality.

Overall we experienced that designing the UIs allowed us to look at the system from different perspectives, increasing our understanding of the involved components, and leading participants to rethink some functionalities.

ACKNOWLEDGMENTS

We would like to thank SKA Perentie and CREAM teams as well as Max IV Institute for having made this work possible through their contribution to the SKA and Taranta projects.

REFERENCES

[1] Rees, N., "SKA Project Status Update," in [*Proc. 19th Int. Conf. Accel. Large Exp. Phys. Control Syst. (ICALEPCS'23)*], *International Conference on Accelerator and Large Experimental Physics Control Systems*, 1610–1615, JACoW Publishing, Geneva, Switzerland (02 2024).

[2] Bartolini, M., Brederode, L., Deegan, M., Miccolis, M., Rees, N., and Santander-Vela, J., "Scaling agile for the square kilometre array," in [*ICALEPCS 2019*], (2019). paper WEPHA011 https://doi.org/10.18429/JACoW-ICALEPCS2019-WEPHA011.

[3] Scaled Agile, Inc., "SAFe for Lean Enterprises 6.0." https://www.scaledagileframework.com/ (2024).

[4] Alberti, V., Baffa, C., Brajnik, G., Colciago, M., Giani, E., Marotta, G., and Novak, I., "A Lean UX Approach for Developing Effective Monitoring and Control User Interfaces: A Case Study for the SKA CSP. LMC Subsystem," in [*Proc. 19th Int. Conf. Accel. Large Exp. Phys. Control Syst. (ICALEPCS'23)*], *International Conference on Accelerator and Large Experimental Physics Control Systems*, 1650–1656, JACoW Publishing, Geneva, Switzerland (02 2024).

[5] Hampson, G. A., Bunton, J. D., Humphrey, D., Bengston, K. J., Jourjon, G., Bolin, A. B., Chen, Y., Troup, E. R., Babich, G. C., and Aardt, J. C. V., "Square Kilometre Array Low Atomic commercial off-the-shelf correlator and beamformer," *Journal of Astronomical Telescopes, Instruments, and Systems* **8**(1), 011018 (2022).

[6] Gothelf, J. and Seiden, J., [*Lean UX: Designing Great Products with Agile Teams*], O'Reilly Media, Inc. (2021, (third edition)).

[7] https://taranta.readthedocs.io/en/latest/. (Accessed: May 2024).

[8] MAX IV Laboratory, "MAX IV: We make the invisible visible." https://www.maxiv.lu.se/ (2020).

[9] Rosson, M. and Carroll, J., [*Usability Engineering Scenario–based development of human–computer interaction*], Morgan Kaufmann (2002).