THE SKA DISH LOCAL MONITORING AND CONTROL SYSTEM USER INTERFACE

A. Marassi, INAF-Astronomical Observatory of Trieste, Trieste, Italy
M. Brambilla, Politecnico di Milano, Milano, Italy
A. Ingallinera, S. Riggi, C. Trigilio, INAF-Catania Astrophysical Observatory, Catania, Italy
G. Nicotra, INAF IRA, Bologna, Italy
G. Smit, SKA South Africa, Cape Town, South Africa

INTRODUCTION

SKA Dish

The Square Kilometre Array (SKA) project is responsible for developing the SKA Observatory, the world's largest radiotelescope ever built: eventually two arrays of radio antennas - SKA1-Mid and SKA1-Low - will be installed in the South Africa's Karoo region and Western Australia's Murchison Shire, each covering a different range of radio frequencies. In particular SKA1-Mid array will comprise 133 15m diameter dish antennas observing in the 350 MHz-14 GHz range, each locally managed by a Local Monitoring and Control (LMC) system and remotely orchestrated by the SKA Telescope Manager (TM) system.

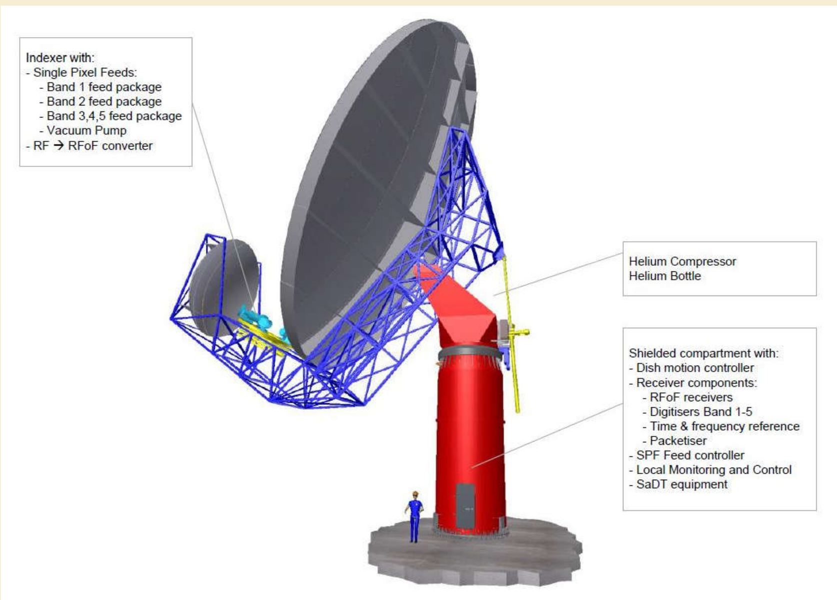

Figure 1 SKA Dish antenna overview

Dish sub-elements

Four sub-elements can be identified in the SKA-Mid1 dish element:

Dish Structure (DS): antenna structure and optics, feed indexer, servo systems, power distribution and safety systems

Single Pixel Feed (SPF): feed packages (OMTs, LNAs, helium cooling and vacuum system and relative controllers

Receiver (SPFRx): RF digitizer and relative controllers

Local Monitoring and Control (LMC): subsystem for each dish antenna that deals with the management, monitoring and control of the operation as orchestrated by the Telescope Manager (TM)

Dish User Interfaces

Two user interface types are assumed from the element side:

Engineering interfaces used by DSH sub-elements engineers for test,diagnostic,maintenance (Figure 2)

Navigation interface (used by control room operators for operations purposes)

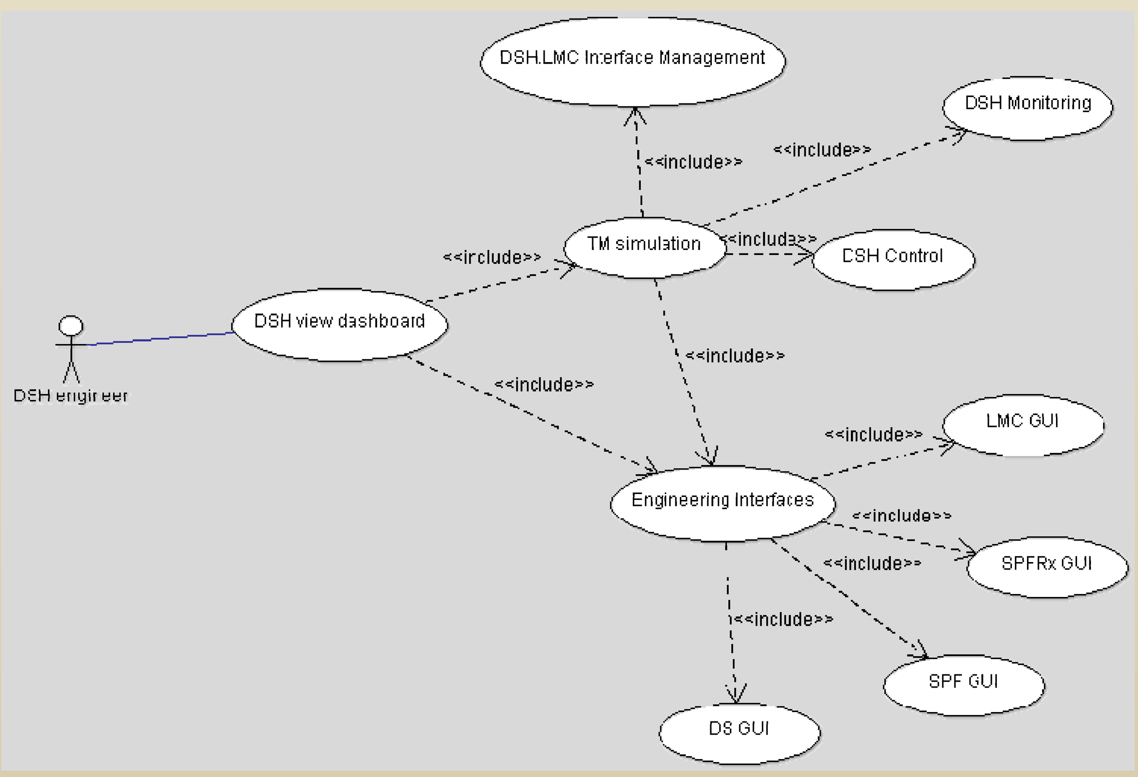

Figure 2 Dish Engineering Interfaces

METHODS

Usability and Accessibility

The *ISO 9241 standard Ergonomics of Human-System Interaction* (ISO, 2008) defines *usability* as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use", specifying:

Effectiveness: the accuracy and completeness with which users achieve specified goals.

Efficiency: the resources expended in relation to the accuracy and completeness of the achieved goals.

Satisfaction: the comfort and acceptability of use

Accessibility is the degree to which a product, device, service, or environment is available to as many people as possible (often focused on people with disabilities or special needs).

USAGE-CENTERED DESIGN

A usage-centered design (UCD) approach[1] for interactive software applications is based on the early involvement of users of the application from its conception. In practical terms, it means that **feedback offered by users** is considered in analysis phases, as well as **iterative** design, prototype and **evaluation** based on **usability** criteria. Several techniques can be applied to establish such a kind of process: structured interviews, contextual enquiries, sketching, storyboarding, user testing, writing scenarios and personas, among others[1,3,4,5,6]

Interactive sketches and storyboards may be used as throw-away mockup UI prototypes (Figure 3) and discussion documents for brainstorming with the aim of eliciting opinions of stakeholders and users.

Figure 3 Sketch of DISH LMC engineering UI

User interaction modeling can also be applied, focusing on expressing the content, user interaction, and control behavior of the UI through visual diagrams that represent the navigation paths of the user. Interactions have been modeled using the standard **Interaction Flow Modeling Language (IFML)**[7]. IFML is instrumental to provide a conceptual view of the user interfaces (see excerpt in Figure 4), which can leads to **automatic verification and quick prototyping** on the target platform of choice through automatic code generators.

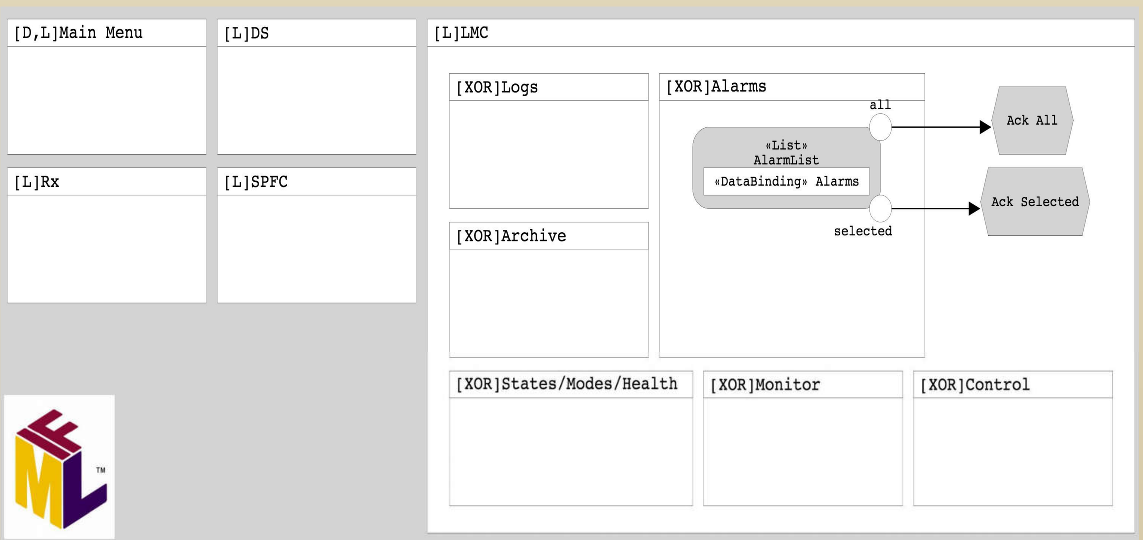

Figure 4 IFML conceptual model of DISH LMC UI

RESULTS

The **Dish engineering Uis UCD pilot project**, although not yet completed, has already given some important hints both from the methodological side and as regards concrete outputs.

It has highlighted the entity of the effort and the time needed to correctly design and implement user interfaces to serve a complex system such as SKA.

It has helped to:

a. elicit new requirements in terms of activities that have to be supported (through techniques such as user analysis, precursors analysis, brainstorming and focus group sessions among stakeholders);
b. study the tasks that SKA users would have to carry out (through task analysis, use case modeling, scenarios definition, sketching and storyboarding);
c. design and validate appropriate UIs (through refinement of sketches, storyboards and low-fidelity prototypes and user testing.

The use of **sketches and storyboards** as discussion documents for brainstorming, with the aim of eliciting opinions of stakeholders, has been appreciated and yielded concrete results. In particular, the use of interactive sketches with clickable hot spots linking to other sketches to simulate Uis, thus illustrating interactively their intended dynamics, has proven to be a valuable instrument to get users' feedback. In this way we explored several design ideas such as the interaction models and the features to implement. The discussion with stakeholders resulted in the definition of a set of user roles and tasks to be performed and in the integration and refining of the initial set of requirements.

As regards the **technological evaluation**, the options for TANGO desktop development (including ATK based on Java Swing, QTango based on C++ and Qt and Taurus based upon Python and PyQt) all fulfill the basic SKA.TM requirements and could be used to implement desktop Uis like SKA.DISH engineering UIs.

IFML [7], an international standard proposed by the **OMG**, has been tested in the user interaction modeling to provide a conceptual view of the Uis. Further steps are envisaged towards quick prototyping on the target platform of choice and automatic verification and validation.

CONCLUSIONS

Proper usage-centered UI analysis and design activities can mitigate product risks (i.e. those concerning with what will be developed and whether it will be the *right* solution), elicit new requirements through users and tasks analysis, design and validate appropriate UIs.

Prototyping is a key tool to be used in exploratory usability investigations and for the evaluation of technologies against SKA TM requirements.

REFERENCES

1. Constantine, L. and Lockwood, L., [Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design], Addison-Wesley Professional, (1999).
2. Greenberg S., Carpendale S., Marquardt N. and Buxton B., [Sketching User Experiences: The Workbook], Morgan Kaufmann, (2011).
3. Rosson M.B. and Carroll J.M., [Usability Engineering: Scenario-Based Development of Human-Computer Interaction], Morgan Kaufmann, (2001).
4. Holzblatt K., Wendell J. and S. Wood, [Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design], Morgan Kaufmann, (2004)
5. Rubin J. and Chiswell D., [Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests], Wiley, (2008).
6. Preece J., Sharp H. and Rogers Y., [Interaction Design: Beyond Human-Computer Interaction], Wiley, (2015).
7. Brambilla M. and Fraternali P. Interaction Flow Modeling Language: Model-driven UI engineering of web and mobile apps with IFML. Morgan Kaufmann, (2014).