ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

USABILITY RECOMMENDATIONS FOR THE SKA CONTROL ROOM OBTAINED BY A USER-CENTRED DESIGN APPROACH

V. Alberti, INAF-OAT, Trieste, Italy
G. Brajnik, Interaction Design Solutions, Udine, Italy

Abstract

User-Centred Design is a powerful approach for designing UIs that match and satisfy users' skills and expectations. Interviews, affinity diagrams, personas, usage scenarios are some of the fundamental tools for gathering and analysing relevant information. We applied these techniques to the development of the UI for the control room of the Square Kilometre Array (SKA) telescopes. We interviewed the personnel at two of the SKA precursors, LOFAR and MeerKAT, with the goal of understanding what features satisfy operators' needs, which ones are missing and which ones can be improved.

What was learned includes several usability issues dealing with fragmentation and low cohesiveness of the UIs, some gaps, and an excessive number of user actions needed to achieve certain goals.

Low usability of the UI and the large scale of SKA are two challenges in developing its UI because they affect the extent to which operators can focus on important data, the likelihood of human errors and their consequences. This paper illustrates the followed method, provides examples of some of the artefacts that were produced and describes and motivates the resulting usability recommendations which are specific for SKA.

INTRODUCTION

SKA (Square Kilometre Array) is an international project to build the world's largest radio telescope [1]. The signal coming from hundreds of dishes and thousands of dipolar antennas will be combined using interferometry to reach sensitivity and resolution much higher than today's best radio telescopes. Given the operational costs of having such a system providing high quality scientific data, to maximise the observational success is a challenge that has to be won.

One of the critical aspect is the design of a User Interface (UI) capable of supporting the operators in the difficult task of running the telescope. Without adopting an adequate development process of the UI, complexity and size of SKA are likely to lead to negative performance by the people that have to operate the system and to unsatisfactory key performance indicators such as high error rate, low efficiency, poor quality of scientific data. In the end these would contribute to loss of time, loss of observing opportunities, poor quality of observations, increased operational costs, damages to equipment, safety problems.

For this reason, to identify the features that clearly help the users or those that, instead of helping, reduce their efficiency is crucial. The User-Centred Design (UCD) approach provides a well consolidated method to tackle the problem. A first step is to conduct appropriate analysis to learn about expected users of the UI to be designed, so that it becomes clear what they perceive being the problem that needs to be solved, and what is the physical, social and conceptual context in which the UI will be used.

Following this method we conducted structured interviews to the personnel involved in the operations of the LOFAR (LOw Frequency ARray) and MeerKAT telescopes.

LOFAR is a fully operational telescope consisting of 51 stations of dipolar antennas spread over Europe and operated by the Netherlands Institute for Radio Astronomy in collaboration with international partners [2]. The MeerKAT telescope has been designed to be the largest and most sensitive radio telescope in the Southern Hemisphere until SKA becomes operational. It is currently being built in South Africa and, when fully functioning in 2020 , it will comprise 64 dishes [3]. Even if MeerKAT will be able to produce high quality science on its own, the telescope will be part of SKA since the first SKA phase. Personnel at both telescopes contributes to SKA technology, science and operations activity.

The two telescopes, LOFAR and MeerKAT, differ in the operational status and in the type of receptors: this allowed us to have an overview of the differences in the two systems and the procedures and activities that are carried out during the normal operations and the commissioning phases.

The collected information has been categorised using affinity diagrams and helped the definition of user profiles, usage and interaction scenarios, sketches and storyboards. What was noticed is that some usability issues are present in the UIs at both the telescopes and are related to the low cohesiveness between different tools and to scalability. Moreover, the operators at both telescopes identified as fundamental the ability to rapidly access all the information needed to diagnose a problem and to understand its impact on the observation that is being carried on. At the moment they have to perform several steps in order to have a complete picture of the situation. Projecting these issues to the UI for SKA, given its size and complexity, it likely that an inadequate UI could emerge, that could lead to poor performance by the staff in charge with operations. Our work has the aim of preventing such a situation to occur by providing the designers of SKA UIs with essential information about the end users and the functionalities that support their work.

In the subsequent sections some background on UCD method is given, followed by a description of the conducted interviews and set of usability concerns that emerged from them and can drive recommendation for the design of SKA UIs.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

SQUARE KILOMETRE ARRAY

SKA will be built in two phases the first of which will include two instruments, SKA1 Mid and SKA Low located in radio quiet sites of South Africa and Australia respectively. SKA1 Mid will comprise 197 dishes with a maximum distance between receptors of 150 km and will investigate the sky in the frequency band 350 MHz -14 GHz, producing a total of raw data output of 2 TB/s. SKA1 Low will include 512 stations of 256 dipolar antennas each and will operate in the frequency band 50 MHz - 350 MHz, with a raw data output of 157 TB/s. The number of antennas is expected to grow up to 1 000 000 during the second phase of SKA. Dishes and stations can be grouped to form up to 16 sub-arrays for each telescope which will allow for many observations to be carried out concurrently [1]. Once acquired by the receptors the signal will be transported through fibre optic cables to the Central Signal Processor (CSP) of each telescope and processed to obtain visibilities, pulsar survey candidates and pulsar timing. Data from the correlator will then be transformed into scientific data by the Signal Data Processor (SDP).

A general description of SKA UI framework and a summary of precursor's UIs can be found in [4]. Information about LOFAR and MeerKAT telescopes can be found at the respective websites [2, 3] MeerKAT UIs are described in [5]. The work that has been done for ALMA UIs is very interesting and can be found in [6–8].

UCD THEORETICAL BACKGROUND

User-centred design (UCD) is an approach for developing interactive software that focuses on users of the system, their characteristics, their preferences, their tasks, and the environment and context in which tasks are performed, from the conception of the system to its deployment. The term user-centred design was coined by D. Norman and became widely popular after the publication in 1986 of [9]. Norman's subsequent very successful book further popularised the concept [10]. A slightly different meaning has the term "Usage-centred design", which was coined by Constantine and Lockwood [11] to characterise an approach that centres on usage rather than users. As such it draws attention on those aspects of users that are most relevant to user interface design. In the following with UCD we will refer to this latter meaning.

UCD is an iterative approach that revolves around activities aimed at exploring the design problem and possible solutions, at implementing partial solutions and prototypes, and at evaluating usability and user experience of such solutions. Through iterations, improvements can be found and applied to the product.

A first pier of the UCD approach is that appropriate analysis needs to be carried out to learn about expected users of the UI to be design, so that it becomes clear what is the problem that they perceive in need to be solved, and what is the physical, social and conceptual context in which the UI will be used. By adopting appropriate analysis techniques a UI designer can understand what are the design priorities (for what kind of users are they designing the UI? Where will such a UI be used? How will it be used? Why will it be used? To do what?). Appropriate analysis techniques include:

- **Semi-structured interviews**, in which appropriately selected stakeholders are interviewed 1-to-1, using a very general interview script; the interview aims at collecting specific and design-relevant information, that may include also information about users, their background and context of activity. See [12].
- **Affinity diagrams**, a qualitative analysis technique based on human clustering. A team of designers collaboratively read, discuss and cluster all factoids that are collected during several interviews. Very specific categories of facts are created and recursively clustered themselves, until the whole body of knowledge has been analysed. See [13].
- **User profiling**, a technique aimed at producing descriptions of categories of users that help figuring out what user-related aspects are relevant for design. A role profile includes a description of the context in which the job role is played (workflow, physical environment, social situation, external sources, background, etc.), a description of characteristics of performance of the role (frequency, intensity, duration, complexity, etc.) and design objectives that are important for the role (ease of learning, efficiency, reliability, accuracy, etc.). See [11].
- **Personas**. Personas are fictitious, specific, concrete representations of target users. This technique aims at constructing 1-page descriptions of key personas of the expected user population for the UI that have been assembled in order to highlight characteristics that somehow affect the design of the UI. Once developed, personas are used "to give the voice of users" whenever there is some design discussion, such as when assessing the potential usefulness of a UI feature, or when assigning priorities to different features based on importance of different personas, or when performing usability assessments. See [14].
- **Scenarios**. Scenarios are specific, concrete narrative description of the behaviour of a potential users (or persona) with respect to the UI. "Usage scenarios" describe in abstract terms what such a user aims at, why, what are his/her expectations and priorities, and what is the course of actions that such a user follows and why. "Interaction scenarios" are more specific and make explicit references to components of the UI (for example specific widgets such as date pickers) that are used to accomplish the task. In both cases scenarios are tuned on the user side, not the system side: they make only cursory reference to the system behaviour. The point of scenarios is to highlight difficulties and challenges that are faced by users and possible ways in which the UI can help. See [15].

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

- **Task models and Essential use cases**. An Essential Use Case is a single, discrete, complete, meaningful, and well-defined task of interest to an external user in some specific role or roles in relationship to a system, comprising the user intentions and system responsibilities in the course of accomplishing that task, described in abstract, technology-free, implementation-independent terms using the language of the application domain and of external users in role. An essential use case model describes the structure of goals and subgoals that certain users may have to achieve during their work. Differently than UML use cases, essential use cases avoid committing to specific UI components. See [16].

A second pier of UCD is that appropriate techniques are used to synthesise artefacts that represent solutions. Some of these artefacts rely on sketches and prototypes, others on intermediate models of the UI. The goal is to obtain some sort of materialisation of the desired UI that can be used to make decisions in terms of its utility and usability. A key aspect is that such materialisations need to be cheap and fast to produce and to revise. Important techniques include:

- **Content modelling**. This is a technique introduced in [11] whereby a UI is conceived as a system of "materials" (the information that users need in order to achieve their goals), "tools" (the set of widgets that users need to manipulate and transform materials), "interaction contexts" (the spaces in which tools can be applied to materials) and "workflow" (the order in which users are expected to visit different spaces and do what they need to do). Such models represent what in more concrete forms constitute widgets of the UI, content of the widgets, screens, navigation tools and overall interaction structure of the UI. The major benefit of this technique is that it does not require to figure out the complete and detailed layout of each screen of the UI in order to understand where certain UI components are potentially more useful to be located.
- **Sketching and storyboarding**. A sketch is a low visual-fidelity and static representation of a screen of a UI, often manually drawn. A storyboard is a story (often an interaction scenario) described through a system of interrelated sketches with annotations that explain the content of sketches and their relationships. The power of sketches and storyboards comes from the fact that they evoke meanings, and let readers quickly imagine the underlying system behaviour. As such they are a powerful means to elicit feedback on early design ideas. Furthermore, they are the language by which designers explore the design space in search of the most suitable solution. See [17].
- **Prototyping**. Sometimes a static sketch cannot be used to make appropriate assessments of utility or usability. This often occurs when the variety and quantity of data to be manipulated, or the number of possible manipulations is too large to be handled with static sketches. In these cases more dynamic representations of the UI need to be developed, such as by using HTML and Javascript with canned data.

Finally, the third pier of UCD is related to utility and usability evaluations. This is needed to gather feedback and to be able to decide on the goodness of a prospective solution. Important techniques include:

- **User testing**. This is an empirical technique aimed at collecting feedback from prospective users of the UI. Using the "think aloud" protocol a facilitator asks a study participant to carry out certain tasks using some UI artefacts (such as sketches). Based on the observed behaviour of a few participants (less than 5) the facilitator can understand what are some strengths and some weaknesses of the UI being tested, and then ask the designer to quickly revise the sketch so that a subsequent round of tests can be done. User testing combined with sketching are extremely powerful and cheap techniques to revise a design and improve its utility and usability. See [18].
- **Heuristic evaluations**. This is a technique introduced in [19, 20] based on analytically assessing the extent to which screens of a UI satisfy ten general usability principles (such as visibility of state, visibility of actions, etc.).

ADOPTED METHODS

This section summarises the methods used to conduct the interviews and highlights how the interaction with users is fundamental to understand their mental model of the telescope they operate and of the problem they have to solve.

Semi-Structured Interviews

The first step of our study was to prepare a set of questions to be asked during semi-structured interviews conducted at the LOFAR and MeerKAT during field trips in 2016. We designed the interviews with the goal of:

- characterising the different user roles involved in operating the telescope,
- characterising the kind of relationship among different roles during operations and the responsibilities of each role,
- understanding the procedures followed when problems occur,
- identifying operators' tasks,
- understanding possible desired improvements related to the UIs, and
- identifying possible problems in using the same UI for SKA.

Examples of questions that were asked are:

- What roles can a person in the control room play and what responsibilities this role carries (for example operators or astronomers)?
- What are the main tasks they have to accomplish?
- Is training provided to the operators?
- Which is the general procedure followed by the staff in case of alarms?

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

- Can examples of UI usage be shown?
- Are there parts of the UI that operators would like to change?

As mentioned above, we interviewed people in different roles that are involved in operating the telescope. This includes not only the operators but also, for example, scientists, and software developers who collaborate for the success of observations. We observed them working in their usual environment and interacting with the tools they normally utilise. Normally this helps in identifying situations where usability issues occur. The number of interviewees and their roles is shown in Table 1.

Table 1: Semi-Structured Interviews

Telescope	Interviewee	Roles
LOFAR	7	1 Operator 2 Scientists 1 System Administrator 2 Software Developers 1 Software Support Person
MeerKAT	8	3 Operator 2 Scientists 1 Chief Scientist 1 Software/UI Developer 1 Operation Supervisor

Affinity Diagrams

The large set of paper notes collected during the interviews has been repeatedly clustered by the interviewers in order to divide the notes in small groups, according to similarity of the topic. During this process duplicates and obvious mistakes were discarded and doubts were clarified through discussions between interviewers, see also [4]. Clustering occurred bottom-up, guided by individual notes, rather than top-down, guided by the questions that generated them. This was done in order to let new topics emerge, if the case. In fact, for each cluster new summary notes were created to capture its theme and the final structure was transferred to an electronic mind map for the final touches. A set of general categories, for example roles, procedures, scheduling tools, etc., were defined to gather clusters belonging to similar areas. The number of initial notes, of intermediate clusters and of general categories obtained for each telescope is reported in Table 2.

Table 2: Affinity Diagram

Telescope	Notes	Clusters	Categories
LOFAR	550	155	14
MeerKAT	1460	434	16

Eight categories appeared in the mind maps for both the telescopes: roles, alarms, procedures (to report about the shift, to contact people, to respond to alarms, to plan short-term activities, etc.), scheduling, UIs, logs, scientific data and software development and maintenance. The remaining categories describing LOFAR are mainly related to monitoring processes and to the characteristics of the different tools used for this purpose. On the other hand, the mind map for MeerKAT shows that topics like commissioning activities, testing and failure examples and diagnosing have been explored in more detail during the visit to the telescope.

Among others results, this analysis made possible to draw a concrete description, or profile, of users at the two telescopes (see the definition of “User profiling” given above) to analyse strengths and weaknesses of the adopted UIs. Operators at both the telescopes work in quiet control rooms and are exposed to a high rate of visual inputs that can increase and be associated with audible signals when some problem is detected. Operators’ primary goal is to make sure that all the procedures needed for an observation to be performed successfully are executed and to maintain the telescope in good running conditions. In case an unwanted situation is detected, the operators have to diagnose the problem and contact the staff in charge of solving it by giving them as much detail as possible. This requires operators to be very knowledgeable of the system they are running and to be very keen on problem solving. Their work is characterised by high responsibility tasks and they have to maintain the ability to think and act rationally in high stress situations.

Some of the design objectives that emerged as important to support the operators are:

- Reliability of interaction: The UI should present only relevant information, in a very clear and unambiguous way, as complete as possible (with the option to get more relevant details on demand). Operators will dig down all the details whenever something will sound suspicious; they should not do it just for the sake of making sure that they can trust what they see. Moreover, appropriate feedback should make the user aware if the system is busy processing something or if a process is stuck. This prevents the user sending the same command many times because he/she cannot understand if the process has been triggered or not.
- Completeness: Operators will need to be able to collect all the data that they need to figure out what has happened (situation awareness) or to diagnose some scientific data artefacts. The UI should provide them with ways to be reasonably certain that they have explored all the available information.
- Fault tolerance/protection: Operators’ mistakes or delays in taking the correct action may lead to undesired situations. Appropriate confirmation of actions may be needed to avoid making slips, as well as reversible (undoable) actions.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

USABILITY EVALUATION

Spending several hours with operators and watching them working has been an incredible opportunity to understand the relationship between them and the system.

Because our goal was collecting opinions and capturing the mental models of operators and scientists, we did not perform usability investigations with user testing or heuristic evaluations. Instead, whenever appropriate we adopted a think aloud protocol and asked them to describe what they were doing and what they are trying to accomplish. In most cases interviewees spontaneously highlighted the strengths and weaknesses of the UI they were using as well as what they wished could be improved in future versions of the UI. As mentioned above, an overall problem in both the telescopes reported by the operators was the lack of a rapid and efficient way to access all the information the operators need to diagnose a problem. Processes, devices and observations are perceived as highly intertwined concepts by operators but the UI sometimes failed to provide ways to establish this relationship. This section reports about the main problems that have been mentioned and some considerations on the possible consequences of not taking them into account when designing the UI for SKA.

Fragmentation and Low Cohesiveness

We use the term fragmentation to indicate the lack of integration between different UI components. It can be due to the use of different tools with their own UIs, to different tools and UIs to monitor separate subsystems (network, hardware components, etc.) or to the lack of coordination between the different teams developing different UIs (scheduling UI, correlator UI, etc.). Some examples of fragmentation that were found are:

- The lack of connections between an alarm of antennas and the procedures to follow to solve a reported problem.
- The lack of relationship between the alarm management and the observation management; as a consequence the operator has to mentally figure out what the connections between the alarmed components and affected observations are, and work with two mostly independent UIs.
- The lack of an easy way to tag or copy logs into the reports.
- The presence of more than one not-integrated systems to communicate with other teams, like JIRA and the IRC chat.

In general, the lack of integration between different UIs leads to potential confusion (same concepts might have different names, or same names might be used for different concepts) and to higher mental load (the sequence of steps required by the tasks has to be mostly in the user mind - the UI does not support it and provides no help - and the sequence may differ for different components). This could lead to higher error rates, lower productivity, more frustration, reduced effectiveness. This could be exacerbated for novice users or for users under stress, which for operators is exactly the situation where the highest risks emerge and where operators mostly need a forgiving and supporting UI (see [21] on the paradox of automation).

Fragmentation of the UI can be a problem for SKA because different teams are designing the UIs for subsystems like the scheduler, the correlator, the SDP. Cooperation should be organised in order to avoid having different teams developing tools with similar functionalities but with different features. Initial steps to try and harmonise the look and feel of the UIs have been taken but a concrete strategy is still under study.

Scalability

By scalability we mean the capability of the UI of adapting to the amount of information to be shown and of effectively visualising different scales of the system. The first point implies that appropriate visualisation techniques should be chosen in order to display and interact with large amounts of data, for example the status of 200 dishes or, even worse, the quality of the correlation for each baseline[1]. The second point considers the possibility of providing the operators with the ability of analysing a subsystem without loosing the context to which it belongs, for example to analyse the status of the cooling system of a dish maintaining the information about the dish name and its position in the sub-array. A possible solution to this specific problem has been proposed for example in [8].

The UIs that were in use at LOFAR and MeerKAT when we visited them would fail to cope with both these scalability issues. When developing SKA UIs, designers will have to keep in mind that visualisation techniques should be chosen in order to support the adaptation to a large number of data and to allow for exploring them while keeping the overall information explicit. Attention should be paid also to avoid user overload: a way to cope with it can be to make detailed information available when needed, for example on user request, and contextualised.

Gaps

Gaps are features that could help operators but that were missing in the used UIs. Examples are:

- A tool that would support scientists and operators to perform pattern analysis of raw data plots to identify those that are “suspect” due for example to RFIs (radio frequency interferences) or correlation errors.
- An integrated scheduling tool. The observations to be carried on are often manually inserted in a spreadsheet or in a Google calendar. The operator would take advantages of an integrated scheduling tool that takes into account the status of the system and the priority of the observation in the queue to schedule the best one to be observed next.

[1] Each pair of different receptors form a baseline. The signals from all the baselines in a sub-array are correlated to take advantages of the interferometric technique. For an array consisting of N receptors the number of baselines is $\frac{N(N-1)}{2}$.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

- A direct link between alarms, faults and solving procedures.
- A support to diagnose RFI due to on site activity.
- An integrated tool that helps to verify the position of a source.
- An easy way to find and contact the personnel on duty or in charge to solve a specific issue should be available. Operators have to search paper notes or separate files to find the desired information.

Extendability

Another requirement that has to be taken into account when developing the UIs for SKA is extendability. It refers to the ability of the UI of incorporating a new feature or functionality as the system grows and evolve. Most of the times the changes are not known in advance but the UI should be designed in such a way that it is as extendable as possible. An example is to leave some space for new content to be displayed for a new tab to be added. The UIs currently in use often make it difficult to be extended for instance with new toolbars, new procedures and new actions on devices. This aspect will be important during the whole life of the telescope that is expected to operate for 20 − 50yr . It is also fundamental during the construction phase when the addition of new or different features or updates of the existing ones are likely to be needed. Moreover, given the already envisaged two stages development of SKA, this requirement is particularly important.

CONCLUSIONS

For a system as complex as SKA the development of UI capable of supporting the staff in charge or running the telescope is a real challenge. The UCD approach chosen for the development of the UI for the control room of SKA allowed us to identify four critical aspects to be considered in order to avoid a low usability of the UI. Poor flexibility, fragmentation of the UIs and possible gaps could lead to a high error rate and frustration in the operators with the risk of reducing the operational time and impacting on the costs. If not taken into account the scalability problem will lead to an interface that is not able of visualising the large number of elements and their related data that users will need to manipulate.

To avoid this scenarios, sketches and storyboards, as well as a task model for the operator, have been developed and used as a starting point to tackle with the problem. In the future more interviews with personnel working at other telescopes will be conducted to improve the definition of SKA roles and of the operational context. This will include the creation of personas and of new scenarios, sketches and prototypes to be used to identify important features for the UI and to perform user testing sessions.

ACKNOWLEDGEMENTS

We thank the personnel at LOFAR and MeerKAT for their hospitality and the enthusiastic and thorough support in helping us understand how the telescopes are managed.

REFERENCES

[1] SKA, "Square Kilometre Array", http://skatelescope.org/, 2017. Last viewed: October 2017.

[2] ASTRON, "LOFAR: Quick start guide", http://astron.nl/~moss/lofar/LOFARQuickStartGuide-v1.0.p, August 2017.

[3] SKA, "Africa - SKA Telescope", http://skatelescope.org/africa/, 2017. Last viewed: October 2017.

[4] A. Marassi, G. Brajnik, M. Nicol, V. Alberti, and G. Le Roux, "A user interface framework for the Square Kilometre Array: concepts and responsibilities", In *SPIE Astronomical Telescopes+ Instrumentation*. International Society for Optics and Photonics, 2016.

[5] M. Alberts and F. Joubert, "The MeerKAT graphical user interface technology stack", in *Proc. ICALEPCS'15*, pages 1134–1137, 2015.

[6] E. Pietriga, G. Filippi, L. Véliz, F. del Campo, and J. Ibsen, "A web-based dashboard for the high-level monitoring of ALMA", in *Proc. SPIE*, page 9152 91521B, 2014.

[7] E. Pietriga *et al.*, "Interaction design challenges and solutions for ALMA operations monitoring and control", In *Proc. SPIE*, page 8451, 2012.

[8] J.Schwarz, E. Pietriga, M. Schilling, and P. Grosbol, "Goodbye to WIMPs: A scalable interface for ALMA operations", in *Proc. Astronomical Data Analysis Software and Systems XX ASP Conference Vol 442*, pages 247–250, 2011.

[9] D. A. Norman and S. Draper, *User-Centered System Design: New Perspectives on Human-Computer Interaction*, CRC Press, 1986.

[10] D. Norman, *The Psychology of Everyday Things*, Basic Books, 1988.

[11] L. Constantine and L. Lockwood, *Software for use: a practical guide to the models and methods of usage-centered design*, Addison-Wesley, 1999.

[12] K. Holzblatt, J.B. Wendell, and S. Wood, *Rapid Contextual Design*, Morgan Kaufmann, 2005.

[13] A. Lucero, "Using affinity diagrams to evaluate interactive prototypes", in *Proc. INTERACT 2015*, Lecture Notes in Computer Science 9297, pages 231–248. Springer, 2015.

[14] T. Adlin and Pruitt J, *The Essential Persona Lifecycle*. Morgan Kaufmann, 2010.

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

[16] L. Constantine and L. Lockwood, "Structure and style in use cases for user interface design", In M. van Harmelen, editor, *Object-Modeling and User Interface Design*, Addison-Wesley, 2001.

[17] B. Buxton, *User Experience: Getting the Design Right and the Right Design*, Morgan Kaufmann, 2007.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THAPL03

[18] J. Rubin and D. Chisnell, *Handbook of Usability Testing*, Wiley, second edition, 2008.

[19] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces", in *Proc. ACM CHI'90 Conf.*, pages 249–256, Seattle, WA, April 1990.

[20] J. Nielsen, "Heuristic evaluation", in J. Nielsen and R.L. Mack, editors, *Usability Inspection Methods*. John Wiley and Sons, 1994.

[21] L. Bainbridge, "Ironies of automation", *Automatica*, 19(6):775–779, 1983.