

UI style guide

Feature: <https://jira.skatelescope.org/browse/SP-1699>

Giorgio Brajnik and Valentina Alberti, Aug 2021

Introduction

Purpose

These design guidelines aim at improving usability of the User Interfaces (UI, or Human-Machine-Interface, HMI) that are expected to be used by engineers who build/debug/verify devices while implementing the SKA telescopes.

Guidelines help dashboard developers when designing the UIs so that some usability and UX (User eXperience) mistakes are avoided.

These guidelines have been synthesised from our own experience, from established usability and UX principles, mostly from those listed in [Alberti-Brajnik, 2018], [Rockwell, 2019], [BSEN-63303, 2021]. The SKA Brand Book [SKAO, 2021] was also considered but whenever its suggestions would conflict with established usability or design principles, the latter would prevail.

We also considered what is possible and not possible to implement today with Taranta. Occasionally we noted where Taranta improvements would be needed in order to fully comply with a specific guideline.

These guidelines are expected to change rapidly, co-evolving with Taranta and with the complexity of dashboards themselves.

When mentioning Taranta we refer to version 1.1.2 and to running UIs (or dashboards) created by users to interact with the system.

In general a dashboard failing these guidelines leads to poorer usability, which reflects in users being confused, frustrated, less productive and less effective than what could happen otherwise.

At the moment there is no distinction between normative and advisory guidelines. A normative guideline mandates a particular property and a dashboard failing that guideline should be considered invalid, unacceptable. Advisory guidelines are suggestions on how to improve dashboards; developers are expected to follow them whenever possible and economically feasible.

Currently, all the guidelines are advisory. In future versions of these guidelines we will introduce normative ones.

In the future there will be at least three different resources that developers could exploit. These will define the "SKA UX Standards" which, according to [BSEN-63303, 2021] include:

- the **HMI philosophy** document, which provides underlying basic guiding principles for the HMI design at our facility - SKA.

- the **HMI style guide document**, which gives implementation guidelines based on the principles described in the philosophy document.
- the **HMI toolkit**, a set of graphical symbols, widgets, colour themes, etc. that are needed to support what is defined in the other documents.

Currently we don't have these resources, but [Alberti-Brajnik, 2018] can be considered as a preliminary version of the HMI Philosophy document; and Taranta itself works as a container of the HMI Toolkit.

In the future, guidelines will also have to address the design of consoles (number and layout of physical screens, of input/output modalities and devices). At the moment we assume that dashboards are shown on the user PC/laptop screen.

Scope

These guidelines apply only to running dashboards that are developed (and served) through Taranta. As said above, the scope is engineering UIs: those used by engineers and supporting tasks like inspecting, monitoring, verifying behavior of Tango devices.

For the moment, given its complexity and the fact that alarm management is still under design, these guidelines **do not** cover the topic of how to develop dashboards specifically aimed at handling alarms. In future versions they will cover that as well.

We assume that dashboards are run on the laptop/PC of the user and that the user interacts by using keyboard and mouse or other tracking device. In particular guidelines **do not** consider touch-based interactions and mobile devices (like tablets and smartphones). Taranta dashboards do not support a responsive nor liquid design.

Engineering UIs

For the time being, Taranta dashboards are developed and used in the relatively restricted context of SKA engineering UIs. However there might be quite a large variety of users and therefore user behavior, expectations and preferences. Developers need to consider them, and doing so prior to implementation is much more effective. Discussion of the design process is outside the scope of this document (details can be found in [Alberti-Brajnik, 2018]), but we would like to draw attention to these differences.

There are at least three different types of users of dashboards, and corresponding usage profiles:

- **developers** of the system. They develop Tango devices and the dashboards themselves. These dashboards might be short-lived (a few weeks) and serve the purpose of supporting inspection, debug, verification of devices; sometimes for demos. Their audience is limited to developers, who are extremely familiar with dashboards and devices. Usage is frequent for a limited time. Bugs or defects in dashboards are not

important (they are either fixed if they impede the user goal, or left there if there is a workaround or they can be tolerated).

- **Other developers.** The dashboard and the devices have been developed by colleagues of these users (who might be testers or other developers not directly involved in the development of dashboards). They use the dashboards occasionally to get a glimpse of the behavior or structure of devices that they are not 100% familiar with. Usage is infrequent and for a limited time. Dashboard defects might have a deeper effect, in that the user ceases to use the dashboard and quits the goal.
- **Managers and other engineers,** like Product and Feature Owners, architects, AIV engineers, members of other teams. They have not developed the dashboards nor the devices. Dashboards are long-lived, they are used for demos, they can be reused many times by different people.

AIV engineers might use a dashboard very frequently and intensively during a testing session, whereas a PO/FO could use it occasionally for a demo. In any case these users have poor familiarity with the underlying devices and the dashboard implementation. Dashboard defects have an important negative effect on these users. Notice that in many cases these users need also to upload specific configuration strings whose default values need to be made available somewhere.

General layout guidelines

The layout of a dashboard will depend on expected screen size as well as on its intended usage which in general can be understood by approaching the problem from the Lean UX perspective (see more details in SAFe pages as well as [Alberti-Brajnik, 2020], [Alberti-Brajnik, 2018]).

Headers and footers

For the sake of dashboard consistency across the whole SKA project (at least for long-lived dashboards) and for branding consider creating dashboards with a header and possibly a footer.

The header should contain at least the approved SKA logo and a dashboard title. The usage of SKA logo is subject to restrictions that can be found in the SKA Brand Book [SKA-brandbook, 2021]. The dashboard title provides context helpful to infrequent users.

The footer marks the bottom of the dashboard and might contain secondary information.

Guidelines are:

- **GUIDELINE:** Consider defining a dashboard layout that has a header and a footer.
 - When deciding if to add a header/footer, consider that the user might be unfamiliar with the dashboard and its devices, that a screenshot of a dashboard can be circulated and also end up in a paper or in a post.

- **GUIDELINE:** Consider adding an explicit title to the dashboard. The title should identify the main function of the dashboard or the main devices that can be supervised with the dashboard
- **GUIDELINE:** Consider using the SKA logo; if so use only the approved logo with its specified colours and sizes.
-

The SKA Brand Book suggests certain colours and colour combinations that can be used for headers and footers, see the table below.

Because some of the suggested colours conflict with other design principles, we suggest being extremely conservative on using them and prefer gray-scale colours or the blueshift Navy colour as a header/footer background. Avoid using the magenta and yellow colours because they are often associated with alarms. See Section “Colours” for more detailed guidelines on their use.

- **GUIDELINE:** If background/foreground colours are needed for headers and footers, use gray-scales.
- **GUIDELINE:** If different hues are needed for background/foreground colours of headers and footers, use only the suggested Blueshift Navy colour.

At the moment Taranta, when running a dashboard, presents a header containing contextual information (e.g. current user, Taranta version, title of the dashboard, current notifications), and means to switch between a view of the dashboard or of devices. No footer is shown. Taranta does not provide widgets for defining and adding custom headers/footers.

To create a header/footer with the current version of Taranta you’ll have to use the Label widget. A header containing the logo and the dashboard title can be obtained following these steps: drag and drop the Label widget on the canvas, change its size and background colour appropriately. Add a second Label widget with the dashboard title text and position it where decided. Add a third Label widget and use the custom css field to add the logo as a background image by using the image url.

Table 1: SKAO official colour schema applied to a general UI layout.

UI element	Colour name	Value RGB HEX	Additional info
Header / Footer background	Blueshift Navy	R7 G0 B104 #070068	If necessary, use Blueshift navy as a header / footer background colour.
Floating action buttons	Redshift Magenta	R228 G7 B105 #E50869	Avoid using this magenta because red and yellow are often used for alarms

Header / Footer text	White or Blueshift Navy	R255 G255 B255 #FFFFFF Alternatively: R7 G10 B104 #070068	Use white for text on a Blueshift Navy background. Use the alternative colour for text on a white background
Header / Footer Icons	White or Blueshift Navy	R255 G255 B255 #FFFFFF Alternatively: R7 G10 B104 #070068	Use white for icons on a Blueshift Navy background. Use the alternative colour for icons on a white background
Dropdown menu text	Blueshift Navy	R7 G10 B104 #070068	if necessary

It would be good if Taranta allowed customization of its default header and addition of custom headers/footers to provide extra information to users. In general headers could include some of these elements:

- The SKA logo.
- A navigation menu.
- The title of the visualised dashboard.
- The date and time in different time systems if needed.
- A search text field if needed.
- A tool for changing visualisation options of the current window (colour theme, units).
- A notification icon indicating whether there are notifications to the user waiting to be seen.
- A user area (perhaps identified by an icon) indicating logged-in user and giving access to user profile settings and logout action.

Likewise footers could include:

- The version of the current dashboard (if versioned).
- Its authors (if relevant).

Positioning interactors on the screen

An “interactor” is any element of the UI that the user can act upon, by clicking, tabbing, dragging, typing. Examples of interactors are links, buttons, radio buttons, checkboxes, text fields, comboboxes, draggable corners or columns.

Fitts law is a time-honored usability principle that can be used to help us understand how close and how large should the targets of interactors be when used with the mouse. The smaller the interactor area, the longer it will take for the user to hit the target and the higher the number of clicking mistakes

(missing the target) made by the user will be. The farther the interactor is from the starting position of the mouse pointer and the longer it will take for the pointer to hit it.

- **GUIDELINE:** Avoid using a small area for an interactor that is to be used frequently. Make it bigger.
- **GUIDELINE:** Avoid putting close to each other two small interactors when the consequences of hitting the wrong one are important. Separate them with white space.
- **GUIDELINE:** If two interactors are expected to be used one after the other, place them relatively close to each other.

Consider also the fact that sometimes the dashboard will be used with UI interaction devices that are not exactly the same as those of the designer, or in a slightly different posture (for example while standing and giving a public demo in a f2f meeting). Under these circumstances the user will have less control on the pointer movement. Adhering to these guidelines will improve usability (and accessibility as well).

Keyboard interaction

Certain users in certain situations might prefer to interact with a dashboard using the keyboard, rather than using only the mouse/trackpad. This is usually done in order to speed up interaction, since the hand does not need to move between keyboard and mouse. For example we can assume that a developer might want to quickly activate different interactors (like buttons, text fields, turning on/off variables of a widget, switching between modes of the timeline widget) when debugging a device.

Interaction happens when the specific interactor gets the "focus" (explicitly, with a keyboard tab or implicitly, by a mouse click or sometimes because the UI activates it).

To support keyboard interaction a running dashboard needs to handle keystrokes like tab, shift-tab, space, enter, esc, left/right/up/down, shift-left/right/up/down, ctrl-left/right/up/down in order to move the focus between "adjacent" interactors, to reveal drop-down menus, to scan entries of menus, to hide them, to activate a function and to select some of their entries.

Relevant guidelines are:

- **GUIDELINE:** Make sure that any dashboard behavior that can be achieved with the mouse can also be obtained through the keyboard alone.
- **GUIDELINE:** Make sure that the tabbing order (i.e. the order in which the focus moves from interactor to interactor) matches the visual order of the interactors.
- **GUIDELINE:** Make sure that when the focus is moved onto an interactor, there are visual cues indicating that. Usually the web browser highlights the interactor that has the focus with a tiny border. However, in many cases this default rendering is so thin that it can be barely noticed.

Taranta partially supports keyboard navigation in running dashboards. Tabbing order between widgets is possible but it is affected by the order in which they are laid out in the canvas and the order of the layers.

Furthermore, Taranta supports only partial keyboard navigation for widgets that embody interactive controls (for example, turning on/off a variable in the attribute plot). Therefore, at the moment, we expect only partial compliance to these guidelines.

Positioning elements on the screen

A UI design needs to add salience to certain elements of the UI in order to increase the speed at which the user perceives and interprets their informational content. Not everything in a dashboard is equally important, and a designer needs to make these differences easily perceivable.

Likewise, elements in a dashboard may be related to each other. It is important that the design makes such a property easily perceivable.

This can be achieved in various ways, for example by the means of size, enclosure, shading/colouring and placement (See [Alberti-Brajnik, 2018] Section 3.4).

In addition to using colours (discussed in a next section), designers should try to use position and size to capture salience. You can follow these guidelines to establish a visual hierarchy (visual cues suggesting what is more or less important):

- **GUIDELINE:** Use a different position and size of an element to make it perceptually salient with respect to others.
- **GUIDELINE:** Group related elements by placing them close to each other, and avoid placing unrelated things close to each other: proximity implies a relationship.
- **GUIDELINE:** Alternatively, group related elements by placing a border around them or defining a background behind them. Elements sharing a common area with a defined boundary are perceived as part of the same group.
- **GUIDELINE:** Respect horizontal/vertical alignments of the enclosing box of a group of elements. Use space as a separator to visually indicate that things are unrelated. Every element should have a visual connection with something else on the dashboard.
- **GUIDELINE:** Use a bounding box with visible borders to identify an enclosure with closely related elements.

Moderation is key: keep emphasis to a minimum. Consider that each perceivable difference may make a difference on how the user perceives and understands the dashboard. Choose the differences you want to achieve.

What is important in a specific dashboard and which elements are related to each other are very general questions that can be answered only after considering the mental model of the user and the structure of the task.

Finally, the user might expect certain information/widgets to be located in certain areas of the screen. If an important part of the dashboard happens to be displayed "beyond the fold" (i.e. below the bottom of the currently displayed section of the dashboard) and the user is not expecting that information, s/he will not look for it and therefore will not scroll down the dashboard, and will miss that information.

Therefore:

- **GUIDELINE:** make sure that the most important part of the dashboard is shown close to the top of the dashboard, in such a way that it does not require a scroll-down action to reveal it.

Interaction

As mentioned above, these guidelines assume that the user interacts with dashboards that are displayed on a PC/laptop by using a mouse (or other pointer-based device) and a keyboard.

In many cases user actions will be performed by moving the pointer and acting with it on interactors made available by the UI.

For the sake of avoiding user frustration, errors and inefficiency, those widgets that should not be used in a certain situation, should be clearly presented as disabled and they should not be operable (dimmed, grayed out, with a different shape of the mouse pointer, not tabbable via the keyboard). In this way, users would quickly understand what controls of the dashboard cannot be used under those circumstances. This understanding should happen without the user having to try to interact with the disabled widget.

Relevant guidelines:

- **GUIDELINE:** Make sure that the presentation of interactors makes it clear that they can be acted upon, if the widgets they belong to are active.
- **GUIDELINE:** On the other hand, when a widget should not be used, make it visually clear that it is disabled so that users promptly see that.

At the moment, when using Taranta the user can identify actionable interactions and distinguish between enabled and disabled buttons for some of the widgets; Taranta uses different shapes of the pointer and different colours to highlight this information. The enabled/disabled status of a button or text field cannot be directly driven by Tango values.

Density of displayed information

Depending on the expected usage profile, the informational density of a UI can change. Users that are trained and that have to supervise a complex process/device might prefer a denser UI with everything under control without having to move the gaze on to different displays or to open up secondary panels or windows. The same UI can be overwhelming or confusing to users that are

less trained or that have a different goal (such as debugging a specific part of the process/device). For Taranta dashboards, density refers to the number of widgets of a dashboard that has a certain size (expressed in pixels).

These guidelines might help in determining how dense a dashboard should be:

- **GUIDELINE:** For dashboards expected to be used frequently by the same person maximize their density, so that they provide all the information and controls that are needed without requiring extra movements of the eyes, head and hands/pointer.
- **GUIDELINE:** Consider that high density means potential clutter, which reduces the ease with which a user identifies the relevant data. In these cases:
 - Reduce confusion by adding salience to important things and establishing a visual hierarchy.
 - Consider removing less important elements, and move them to secondary dashboards to be opened on-demand.
 - In any case pay attention to alignment and space between elements: both should be used to support identification of the elements.
- **GUIDELINE:** Avoid small density dashboards (i.e. those with a lot of unused space). Users will have to spend resources to scan the entire dashboard to find the information. Concentrate the visual elements in a particular section of the dashboard and make it just large enough to contain them.

In the current implementation of Taranta the minimum size of widgets is a constraint and is relatively large. This impacts the ability of designers to create compact dashboards.

Design elements guidelines

Fonts and typography

There are several factors in typography that affect legibility (recognizing letters) and readability (recognizing words); they include font family, text size and text style.

Currently Taranta supports Helvetica and Monospaced Courier New fonts for most of the widgets, for some also Verdana can be used. It is envisaged that the list of available fonts will increase in the future, to include at least those suggested by the SKA Brand Book (Noto and Verdana as a secondary choice).

Suggested guidelines:

- **GUIDELINE:** Choose a font-family and stick to it in your UI to avoid creating differences that don't matter or that can be hardly detected.
- **GUIDELINE:** Introduce differences in text size only when necessary to create visual hierarchy.
- **GUIDELINE:** Use a text size that is appropriate for the assumed visual distance and vision capability of users. An effective and easy test is: while looking at your UI, partially close your eyes to reduce the amount of light

being perceived. You should be able to read the most important words like titles, labels, values.

- **GUIDELINE:** Avoid using the *italics* font style, because it reduces readability and the contrast it creates is poor.
- **GUIDELINE:** Avoid underlining words, because users might associate them to clickable links.
- **GUIDELINE:** Use **boldface** as a visual cue to create salience.
- **GUIDELINE:** If you decide to use text style and size to create contrast, then make elements **really different** by creating big differences.

The current suggestion is to maintain Taranta default settings and use Helvetica to assure consistency across widgets.

Dashboards often include static or dynamic text. Static text is text that doesn't change while using the UI - as opposed to text being displayed as the value of an attribute of a device, for example, which is updated dynamically. Examples of static text are labels.

The following guidelines apply to static text [BSEN-63303, 2021]:

- **GUIDELINE:** Acronyms and abbreviations should be avoided unless they are part of the established user language.
- **GUIDELINE:** If a paragraph is needed, for example when displaying additional info or when showing error messages, avoid uppercase lettering in favour of capitalized text to improve readability.
- **GUIDELINE:** Orientation of text should follow the normal user reading direction (left-to-right for UIs in English).
- **GUIDELINE:** Blinking and moving text should be avoided.
- **GUIDELINE:** Use black as a text colour on a white background as a default choice. Do not reduce the default opacity of 100%.
- **GUIDELINE:** Justification of text should be consistent. For labels make sure that proper alignment is maintained to facilitate comparison of values and comprehension. In particular:
 - Labels for text fields should be placed to the left or above the field and should be left-justified.
 - Labels for checkboxes and radio-buttons should be placed to the right of the control and left-justified.

Dynamic text is text that is provided by the monitored device, and can change frequently over time. For dynamic text the guidelines are:

- **GUIDELINE:** Orientation of text should follow the normal user reading direction (left-to-right for UIs in English).
- **GUIDELINE:** Consider an appropriate size of the area in the dashboard that can be occupied by these values. Make sure that the most frequent values can be displayed without being cropped and without addition of horizontal and vertical scrollbar in the text field.
- **GUIDELINE:** If this is not possible, then make sure that the initial part of the text provides a clue of the rest so that the user can guess it without having to scroll it to the right.

- **GUIDELINE:** Consider using a monospaced style for representing text values so that alignment can be kept.
- **GUIDELINE:** Use black as a text colour on a white background as a default choice. Do not reduce the default opacity of 100%.
- **GUIDELINE:** Consider representing dynamic text with typographic choices or colours that make it stand out from static text.

Numeric values

Numbers used in SKA dashboards can represent values of a range of different variables such as frequency ranges, pointing coordinates, voltages and so on. Their visual presentation should be consistent across the UI and the size and font should be such that they can be distinguished from the user's normal working position.

When presenting a number the user should be immediately able to understand its meaning, without the need to perform conversions between different units of measurements [BSEN-63303, 2021]. A reference list of the physical and engineering units should be available for consultation when needed.

Only the level of precision needed by the user should be displayed (it is better to avoid showing 5 decimals if 3 are sufficient to perform the user's tasks).

The guidelines are:

- **GUIDELINE:** Negative numbers should be preceded by a - sign.
- **GUIDELINE:** When labels are needed for a numerical value, they should be above or to the left of the displayed number.
- **GUIDELINE:** Unit of measure should be present unless made obvious by the context or the value is dimensionless. They should appear at the right side of the number.
- **GUIDELINE:** Numbers should be right-justified so that decimal points are aligned.
- **GUIDELINE:** Integers should not have a decimal point.
- **GUIDELINE:** Leading zeros should be displayed only for numbers between -1 and 1.
- **GUIDELINE:** Ranges should be indicated using the "a-b" notation .
- **GUIDELINE:** Use a presentation standard (labels, units, typography) for numbers that is consistent through the UI.
- **GUIDELINE:** Use scientific notation when appropriate.

Colours

Colour differences are processed very quickly by our visual system and they can be used to effectively drive the user's attention to a specific area of the UI [Alberti-Brajnik, 2018]. If used inappropriately, not to distinguish or convey information but for other purposes, colours can distract rather than support the user in focusing on important information. Moreover, people suffering from vision impairments may not be able to distinguish colours as well as other people.

The following guidelines apply to colour usage when designing the engineering UI for SKA (see [BSEN-63303, 2021], [Rockwell, 2019], [Alberti-Brajnik, 2018]):

- **GUIDELINE:** Use colour only when they differentiate information or direct the user attention to a specific part of the screen.
- **GUIDELINE:** Prefer grayscale colours for backgrounds. If other hues are needed, avoid using bright background colours for large areas.
- **GUIDELINE:** On the other hand, if small symbols need to be placed on large backgrounds, then use bright saturated colours for the foreground symbols.
- **GUIDELINE:** Limit the use of foreground colours to important information. Assure a luminance contrast of at least 3:1 between the foreground and the background UI elements.
- **GUIDELINE:** Use colours in association with other indicators such as text, size, texture, brightness.
- **GUIDELINE:** Use colours exclusively. Don't use the same colour with different meanings.
- **GUIDELINE:** Prefer light mode as default setting. Even though the ability to switch to a dark theme may be available, studies showed that users perform better (higher ability to focus on details) when using light mode than they do when in dark mode.

The following table defines a colour schema that can be adopted when creating backgrounds for the engineering UIs of SKA.

Table 2: Proposed colour set for visual grouping elements.

UI element purpose	Colour name	Value RGB HEX	Additional info
Background 1	Light gray 245	R245 G245 B455 #F5F5F5	As an alternative use white with a border: R255 G255 B255 #FFFFFF
Background 2	Light Gray 222	R222 G222 B222 #DEDEDE	
Background 3 ¹	Dark Gray 117	R117 G117 B117 #595959	
Box border	Light Gray 163	R163 G163 B163 #A3A3A3	

¹If the desired contrast can't be reached due to limitations in the configuration of command widgets, consider using a darker background colour such as Dark Gray 67, R67 G67 B67 | #434343.

Luminance contrast

Luminance contrast is the difference in the amount of light between two objects positioned next to each other [Alberti-Brajnik, 2018]. When the contrast ratio is too low the readability of the foreground element is often affected negatively and consequently also the user experience. Luminance contrast is opposed to hue contrast which is related to the different hues used by adjacent elements. The human eye is about 3 times more sensitive to luminance contrast.

On the other hand, when used appropriately, brightness contrast can be very effective as a highlighting method.

When designing a UI consider the following:

- **GUIDELINE:** To create an effective visual contrast, exploit luminance contrast. Use hue contrast only as an additional contrast mechanism if needed.
- **GUIDELINE:** Adding a small contour to objects helps in distinguishing them from the background.
- **GUIDELINE:** Using an adequate luminance contrast is crucial in supporting people with visual impairments, including those suffering from colour blindness.

The Web Content Accessibility Guidelines report the following values for luminance contrast of different elements of the UI [WCAG 2.1]:

Standard Version	UI element	Contrast ratio
WCAG 2.1 level AA	Normal text	4.5:1
WCAG 2.1 level AA	Large text	3:1
WCAG 2.1 Level AAA	Normal text	7:1
WCAG 2.1 Level AAA	Large text	4.5:1

As a guideline consider to:

- **GUIDELINE:** Make sure that background/foreground elements have a luminance contrast as required by WCAG 2.1 AA.

Amongst others, this tool can be used to check appropriate contrast levels:

- <https://webaim.org/resources/contrastchecker/>
- <https://juicystudio.com/article/colour-contrast-analyser-firefox-extension.php>

Alternatively, printing the UI in black and white is a very effective way to assure that the contrast level is appropriate.

Compliance with this guideline may be very important when defining UIs for alarm handling: if a background is present, make sure to choose a colour that

ensures enough contrast with foreground ones, for the different alarm conditions.



Figure 1: Effect of brightness contrast on readability. The contrast ratio between the text and the background is 1.8:1 in the image on the left, while it is equal to 7:1 for the image on the right. As a result, a white text on a dark background is much easier to read than the same text on a light background.

Icons

An icon is a graphical element that represents something. They are processed by our perceptive mechanisms differently than text, and therefore can be used to support faster usage of a dashboard. For example the same icon can be used on different visual elements to provide cues that those elements are related. Or, vice versa, two different elements can have different icons to make it clear that they differ.. Different types of icons can appear on a UI: system icons, normally used for folders, common actions, system settings and project-related icons that are more specific to the working context (i.e. icons representing an antenna or weather conditions, alarm conditions). General guidelines for icons are:

- **GUIDELINE:** Use icons when necessary to support conveying the information to the user.
- **GUIDELINE:** Because interpretation of symbols is based on previous experience and shared understanding, use well known symbols and add explanatory text especially when introducing new icons. If you use tooltips to provide such information, bear in mind that actions such as mouse hovering increase the cost of interaction.
- **GUIDELINE:** Use the colour palette defined for alarm for those icons that are used to represent faulty conditions or problems.
- **GUIDELINE:** Avoid shuffling icons between different screens. Users tend to associate the meaning of icons to their position on the screen.
- **GUIDELINE:** Limit the number of icons to the essential ones.

A set of Material Design icons can be found at [MD-link].

With the current version of Taranta, icons can only be added as background images to the label and command-file widgets using a CSS rule to specify the image url.

Buttons

Buttons are used to trigger actions on the system.

Relevant guidelines are:

- **GUIDELINE:** Keep the style of buttons (size, colours, words) consistent across the UI.
- **GUIDELINE:** Label buttons effectively. Choose a verb that clearly describes the action that will be performed. If more than one word is required, keep the label as short as possible.
- **GUIDELINE:** Avoid adding borders if they are not necessary.

- **GUIDELINE:** If the action triggered by pushing the button is destructive (i.e. it might lead to waste of users work), make that explicit by means for example of the background colour of the button or label.
- **GUIDELINE:** Avoid using buttons if a link can provide the same functionality.
- **GUIDELINE:** Position buttons close to the related components so that the user knows she is clicking on the right thing.

Designing systems of interconnected dashboards

For systems made of large aggregates of devices it makes sense to design several dashboards that are interconnected, in the sense that a given dashboard can provide ways to open other dashboards that are somehow related to the former.

The following paragraphs provide suggestions that can help making decisions on what dashboards to design and how to spread information and controls on them. These paragraphs do not contain (yet) specific guidelines, but we believe they can be nevertheless useful when deciding how to design dashboards. Furthermore, Taranta at the moment provides little support for managing interconnected dashboards.

The [BSEN-63303, 2021] standard describes four different hierarchical levels of displays. They progressively give access to an increasing level of details making more information available while narrowing the scope of what is presented. The organization of such displays must follow an analysis of the users, their tasks, the context in which those tasks are carried out, their mental model and the workstation² layout.

The four levels are:

Level 1 - Overview Display: this is the display with wider scope, showing the entire span of responsibilities for the user. It gives a high-level overview of the current status of the system, clearly highlighting situations that require immediate attention. No control functionalities should be performed from this display.

Information to be shown at this level are frequently collected in dashboards and contain at least part of the following elements:

- High priority alarms;
- Indications of anomalous situations and of their severity; deviating trends with the direction of the deviation;
- High level performance indicators, key process parameters, key calculated parameters;
- Upstream/downstream unit info;
- Major equipment status.

The primary purpose of this UI is to provide the user with enough contextual information to determine what actions they may need to perform. If different operational modes are allowed, they should have separate dedicated displays

² Workstation: the set of screens and devices used for interacting with the system.

because the set of relevant information that needs to be quickly assimilated may differ.

The definition of the level one interface often follows the creation of the level two displays: for SKA this could be a dashboard showing the overall status and performances of the telescope.

Level 2 - System and subsystem overview displays: they are likely to be the user's primary interface, used when monitoring the system, performing routine changes, and conducting normal operations. They provide overviews of the status of the system and subsystems and should be designed keeping in mind the user mental model of the system and its operations as well as provide easy access to other related displays in the hierarchy. In case different views of the same subsystems are needed different displays can be created to support specific tasks or situations. There may be multiple level two displays for the same equipment to cover specific situations such as startup, normal operation, state or product transitions, and shutdown.

Examples of elements that can be found in level two displays follow:

- Alarms useful in the context - if screen space is a limiting factor visualise only the highest priority ones with an indication that more are present;
- Controls needed to support user's tasks;
- Navigation between different views and other elements in the hierarchy;
- Indicators of abnormal situations, denoting severity.
- Performance indicators for example used bandwidth.

Level 3 - System and subsystem details displays: they are task based displays that contain much more details and allow to control and manipulate items not available from level two displays. If possible, they are configured in such a way that they limit the required navigation to different displays.

Examples of the elements that can compose a level three display are:

- Alarms of all priorities that are relevant to that subsystem;
- Subsystem detailed view - i.e. blueprints with indicators of the status of the subsystem;
- Subsystem controls.

Level 4 - Detail diagnostic display: used intermittently, when detailed information is required, they often don't need a full screen display but can be pop-ups. Examples of layer four displays are:

- Alarm displays with details on a specific sensor;
- Contact information;
- Detailed info about instrumentation;
- Help displays;
- Operating procedures;
- Alarm documentation and response information.
- Systems-supplied displays;
- Shift log information;
- Others.

Process or system control can't be performed from level four displays.

References

[MD-link] <https://materialdesignicons.com/> downloaded 2021

[WCAG 2.1] <https://www.w3.org/TR/WCAG21> downloaded 2021

[Rockwell, 2019] Rockwell Automation Process, HMI Style Guide, PROCES-WP023A-EN-P May 2019.
<https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/proces-wp023-en-p.pdf>

[BSEN-63303, 2021] BS EN 63303 Human-Machine Interfaces for Process Automation Systems, Draft, 12 March 2021

[Alberti-Brajnik, 2018] V. Alberti, G. Brajnik, UI Design Principles, Revision 2 2018-06-29,
http://ska-sdp.org/sites/default/files/attachments/ska-tel-sko-0000787-02_uide_signprinciples.pdf

[SKAO, 2021] SKAO Communication Team,
https://issuu.com/ska_telescope/docs/skao_brand_book, downloaded 2021

[NNG, 2020] Nielsen-Normal-Group, Dark Mode vs. Light Mode: Which Is Better? 2020, <https://www.nngroup.com/articles/dark-mode/>

[Alberti-Brajnik, 2020] A lean UX approach for the engineering user interfaces in the Square Kilometre Array project, Proceedings Volume 11452, Software and Cyberinfrastructure for Astronomy VI; 1145213 (2020)
<https://doi.org/10.1117/12.2562544>