User Interface Design Changes to the Integrated Information Display Concept

Don K. Coady
Scientific Programmer
DRDC – Atlantic Research Centre

Defence Research and Development Canada

Reference Document
DRDC-RDDC-2016-D053
October 2016

Abstract

The Integrated Information Display (IID) is a situational awareness display concept proposed for the command teams of Victoria-class submarines. A functioning prototype of the IID was constructed and trialed as part of a larger Victoria-class simulation. This report describes several changes made to the original IID design as a result of the development and test phases, including the specific User Interface (UI) issues each modification addresses.

Significance to Defence and Security

The Integrated Information Display aims to lighten the cognitive demands of the decision making team within the Victoria-class operations room and thereby increase warfighting capabilities. More specifically, the IID proposes a more readily accessible and integrated source of ownship, environmental, and tactical information to improve situational awareness, and by extension, the quality and timeliness of C2 decisions. The UI design of the IID is critical to its usability, and ultimately to its success as a decision aid. The underlying lessons learned in the design of the IID, including the importance of prototyping and testing iterations to expose specific runtime problems, apply equally to the design of any human computer interface.

Résumé

L'affichage intégré de l'information (AII) est une vitrine sur la connaissance de la situation présentée aux équipes de commandement des sous marins de la classe Victoria. Un prototype fonctionnel d'AII a été construit et mis à l'essai dans le cadre d'une simulation plus vaste se rapportant à la classe Victoria. Le présent rapport décrit plusieurs changements apportés à la conception initiale de l'AII à la suite des phases de l'élaboration et des essais, notamment chacune des modifications liées aux problèmes particuliers de l'interface utilisateur.

Importance pour la défense et la sécurité

Le système d'affichage intégré de l'information a pour but d'alléger les exigences cognitives en matière de prise de décision pour les équipes de la salle des opérations de la classe Victoria et d'augmenter ainsi la capacité de combat. Plus précisément, l'AII propose une source d'information plus facilement accessible et intégrée concernant le navire, l'environnement et les renseignements tactiques. Celle ci permettra des décisions C2 plus éclairées, et ce, en temps opportun. La conception de l'interface utilisateur de l'AII est essentielle à sa convivialité, ainsi qu'à son succès comme aide à la décision. Les leçons tirées de la conception de l'AII, notamment l'importance du prototypage et de la mise à l'essai des itérations en vue d'exposer les problèmes particuliers d'exécution, s'appliquent tout aussi bien à n'importe quelle autre interface homme-machine.

Table of Contents

List of Figures

1 Introduction

1.1 Background

The concept of a new Integrated Information Display (IID) for the command team of Victoria-class submarines was investigated by the Maritime Command Team Support (MCTS) research group at DRDC – Atlantic Research Centre [1]. A working model of the IID design was developed using Adobe Flash and connected to the lab's virtual Victoria submarine simulator [2]. Experimental trials were then conducted using Victoria-class crew members and enabling subsequent evaluations of the IID's potential effectiveness. Usability issues exposed throughout construction and testing of the IID spurred several User Interface (UI) design changes and are the focus of this document.

1.2 General

The original IID design is described generally using an 'area' based layout hierarchy (Figure 1) [1], which has also been used as the basis for this document. Briefly, these UI design areas are as follows:

- Area 1 – Date-Time Group
- Area 2 – Ownship Vitals
- Area 3 – Sound Velocity Profile (SVP)
- Area 4 – Overall Tactical Picture (map)
- Area 5 – Contact Management
  - Bearing View
  - Range View
  - Periscope View
  - Sonar View
  - COI View (contacts of interest)
- Area 6 – Schedule of Events (planner)
- Area 7 – Alert Panel
- Area 8 – Dynamic Content (user defined)
  - Watch List
  - ROE (Rules of Engagement)
  - Totes (contains several sub panels)
  - Weather
  - Events

- Library (contains several sub panels)
- Weapons[1]
- Platform State (not used)
- Contact List

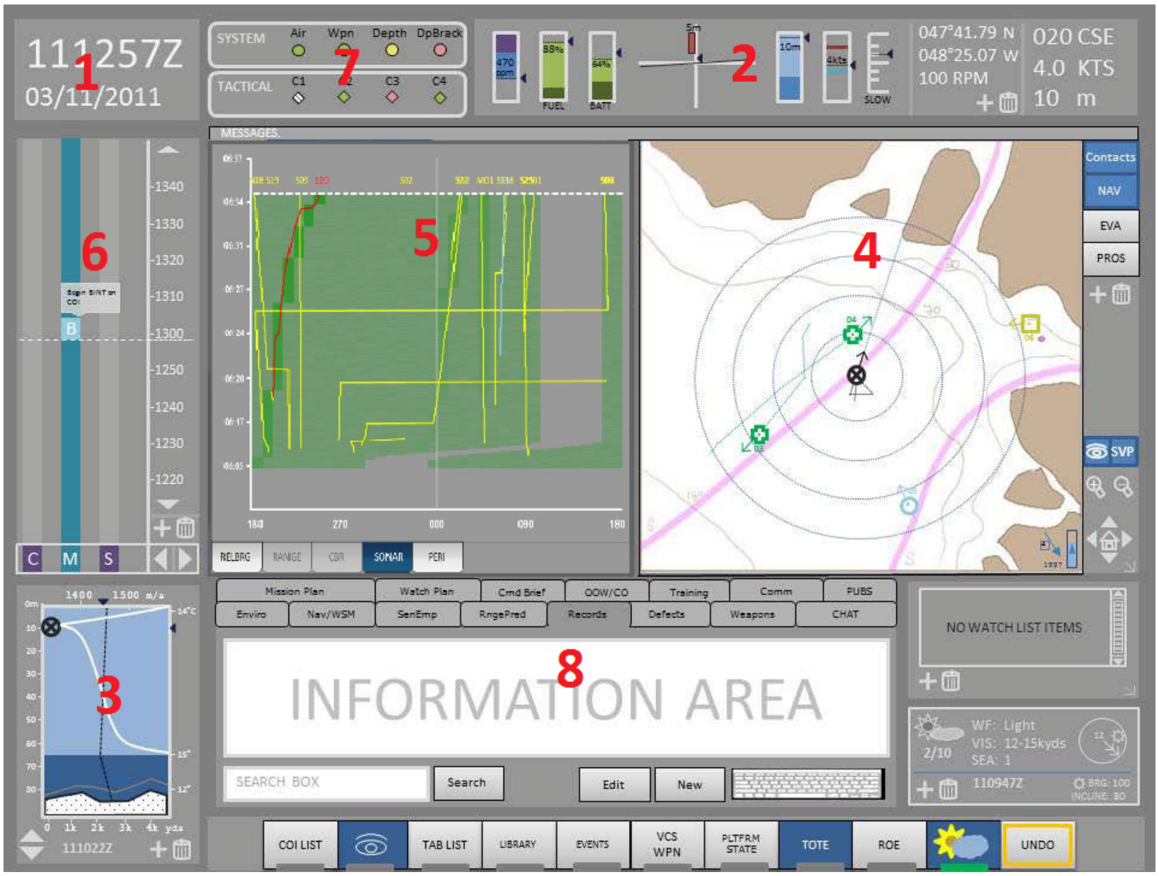

Figure 1: Original Integrated Information Display (IID) design (image modified from [1]).

Areas 5 and 8 utilize tabbed panels (aka ‘viewstacks’) to access their sub-content, i.e., multiple panels and views that are vertically stacked (similar to a deck of cards). Area 8 (bottom region of the display) is unique in that its content is determined by the user ‘on-the-fly’. Any content the user ‘toggles’ on in this area at the bottom of the IID is automatically resized to share this same (container) space.

1.3 UI Terminology Notes

The IID design is a diverse assembly of standard, customized, and completely novel interface components. The complexity of these GUI elements ranges from simple buttons and text readouts

[1] The IID experimental plan did not involve any weapons and as a result the Weapons tote was not implemented.

to advanced charting and data visualizations. Of note within the IID, the various buttons used extensively throughout the interface belong to subtle yet distinct types. Simple, toggle, and tab buttons all share a similar appearance, but each exhibit their own distinct interactions. They are described upfront so that the reader can distinguish between these terminologies as they read through the document.

Generally within traditional interface designs, simple (aka command) buttons are by far the most common and familiar. Recall each click invokes a single consistent action, and repeat clicks simply trigger the exact same action over and over again. Simple buttons cannot persist in a pressed (down) state; they reset to their depressed (up) position after being clicked. Consequently, they cannot store or indicate a system state or setting. The left and right arrow buttons in Figure 2 below are examples of simple buttons. Clicking them will trigger the exact same action (e.g., horizontal scrolling), and will not alter their appearance or state.

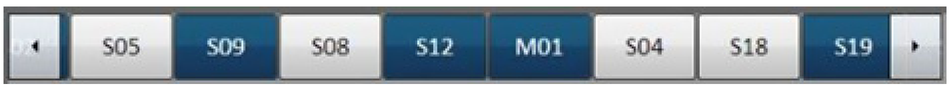

Figure 2: Toggle buttons with simple (arrow) buttons for horizontal scrolling.

In contrast, toggle buttons have two possible states (positions); they can be switched either ‘ON’ or ‘OFF’ (Figure 2). The ON condition is indicated by the ‘pressed’ (down) position of the button and its blue fill color. Importantly within the IID, each button in a group of toggle buttons behaves independently. In other words, pressing any particular toggle button within a group will not toggle off its adjacent ‘sibling’ toggle buttons. This behavior may seem contrary to those familiar with older mechanical toggle arrays in which pressing one toggle button affected (e.g., turned off) all the adjacent buttons.

Despite a similar appearance to toggle buttons, the navigation tab buttons (or tabs for short) have a subtle visual difference (Figure 3), and more importantly, behave quite differently. Pressing any tab button deselects all other sibling tabs, thus only ONE tab within a group can be selected at any given time. Generally speaking, tabs are used for navigating through stacked panels, while toggle buttons are used to switch content (e.g., layers) visibility. Note menu buttons are essentially toggle buttons arranged in functionally related (often vertical) groups. Further details regarding menu design are presented in Section 2.5.

Figure 3: Tab buttons.

Generally within the IID, the intent is for operators to distinguish among the different button types based on visual cues, positioning (e.g., arrangement), and experience.

2 UI Design Modifications

2.1 General Layout

Early in the construction of the IID it became apparent the GUI layout was much more constrained vertically than horizontally. This was partially due to the wide aspect ratio of modern display monitors combined with the square aspect ratio of the larger IID GUI elements, namely the Tactical Map and Contact Management (Areas 4 and 5 of Figure 1). One of the ways this manifests is as narrow constraints on the height of GUI widgets on the top and bottom rows of the IID. For example the upper row of gauges becomes very challenging to construct with legible labels while maintaining their defined shapes. At the same time, as the bottom row (Area 8) narrows, its content becomes less visible and forces excessive vertical scrolling.

The vertical space constraints in Area 8 were further exacerbated by the persistent row of very large toggle buttons (Figure 4) used to control which user determined content panels were displayed.

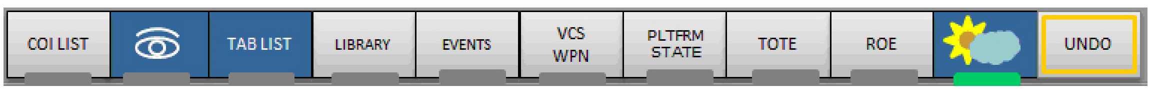

Figure 4: Horizontal oriented toggle button bar with embedded pinning sub-buttons.

Several other issues arose during construction and testing with the original design of the horizontal toggle button bar, and in particular interaction with its ‘pinning’ sub-buttons. These small rectangular grey/green areas inset on the bottom of each button are used to toggle the persistence of its ‘ON’ state and its corresponding display widget. However, usability testing revealed they were difficult to interact with; their narrow size combined with their close proximity to the display bezel made it difficult for users to reliably and accurately touch them. Furthermore, SME discussions revealed confusion with the basic behavior of toggle button groups. At essence of the issue was whether or not turning toggle buttons on was additive. For example, if two toggle buttons are already ‘ON’ and a third is turned on, does this result in three buttons ON or just one (the most recently pressed)? The former behavior was likely assumed for the original design and spurs the need for the nested pinning button. If, however, the behavior of toggle buttons is to stick on until discretely pressed again then the added complexity of the nested pinning button becomes unnecessary. Although there was no clear preference for either behavior, the ‘stick on by default’ behavior was eventually favored and implemented on the basis of simplicity, and more importantly, consistency with similar toggle button groups in other areas of the display.

Another somewhat minor inconsistency identified during the build process was the fact that all except two buttons used text labels. The watch list and weather toggle buttons used symbols unlike their sibling buttons. Again for simplicity and consistency, the symbols were replaced with text labels. A final layout modification in this area was to reorient the toggle bar vertically and essentially transform it into a menu. This alteration freed more vertical space for Area 8’s dynamic content and enabled the addition of a ‘maximum map’ mode for Area 4. This feature allows the operator to toggle between the original sized tactical map of Area 4 and an enlarged map that uses all of Areas 5 and 8 (Figure 5). This capability was added as a direct result of SME feedback and suggestions during usability testing.

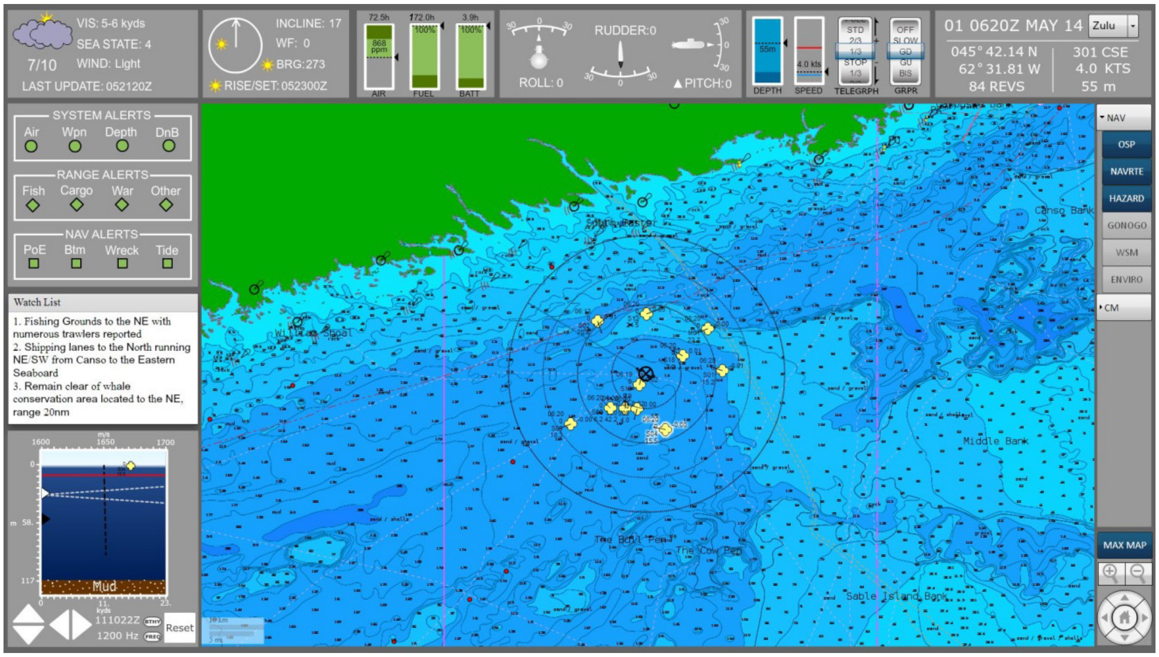

Figure 5: Modified IID design in expanded Area 4 (max map) mode. Note data displayed in this figure is based on a fictitious scenario.

Early usability testing and SME feedback also suggested the Area 6 Event Planner (Figure 6) would be more readable and intuitive if it was oriented horizontally instead of the original vertical design. Many if not most temporal visualizations are constructed on the basis of a horizontal timeline (e.g., Gantt charts). For these reasons the event planner was reoriented horizontally. However, due to the narrow width constraints of its original location on the left side of the IID, the planner was relocated to the much wider dynamic area 8 at the bottom of the display. The vacant space left behind was subsequently used for the redesigned Alerts panel and the Watch List.

Figure 6: Horizontal scrollable planner (left) and vertical toggle button menu (right). Note data displayed in this figure is based on a fictitious scenario.

2.2 Area 1 – Date-Time Group (DTG)

Area 1 was the simplest GUI element within the IID and only a few minor changes were made to the original design. The date-time format was first updated to comply with standard Zulu format (DD HHMM MON YY). The DTG element was then rearranged horizontally to conserve valuable vertical space and also enable it to be merged with ownship vitals in Area 2 (Figure 7). Lastly, an interaction was added to enable the operator to toggle between local and Zulu time display formats. A standard toggle button was initially considered to achieve this; however, this type of UI component can only indicate the current setting, with no (simple) means for indicating the inverse setting. A toggle switch (common on smartphone GUIs) would indicate both the current state (e.g., Zulu) and the available switched state (e.g., local); however, it was simply too wide. A radio button group was also considered; however, it also would not fit given the vertical and horizontal space constraints. Ultimately a combo button (aka drop menu) proved the most spatially condensed solution. The biggest cost to this approach in terms of usability is the operator must tap it twice to switch between the two time settings; firstly to reveal the drop menu, and secondly to make a selection. This trade-off between real estate and added interactions is fundamental to UI design; basically if content cannot be juxtaposed, then added steps are needed to view or navigate to it.

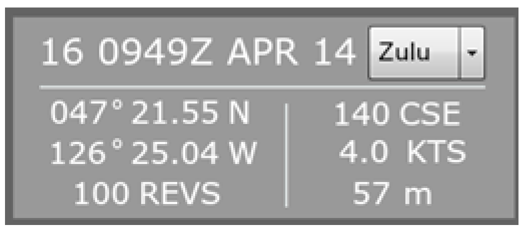

Figure 7: Modified date-time group.

2.3 Area 2 – Ownship Vitals

Other than the text based digital readouts, the solid fill style gauges were the simplest GUI components in Area 2. Despite their seemingly basic graphic design, issues arose early in the build process relating to their dynamic behavior. More specifically, looking at the original design for the fuel gauge as an example (Figure 8 leftmost), the dark green fill at the bottom of the gauge represents the low threshold value. The ambiguity begins when the remaining fuel percentage drops below this critical level. If the light green fill continues to disappear below the low threshold level the dark green fill area will occlude the actual fuel remaining level.

An obvious solution might be to allow the dark green fill to decrease or adjust to actual fuel levels when below the threshold; however, this effectively misrepresents the low threshold setting. Following some brief brainstorming, several alternate behavior designs were considered (Figure 8) of which the design depicted furthest right was chosen. This particular design variation uses red fill for the deficiency gap between the low threshold and actual fuel remaining levels. As a result, the remaining and threshold values for ALL fuel levels are clearly represented. Growing emphasis is placed on the discrepancy gap between the low threshold and the remaining fuel by the increasing amount of red fill (red color is commonly associated with critical levels). Design details aside, this example illustrates the importance of functional prototyping to expose latent usability issues associated with specific system states and use cases.

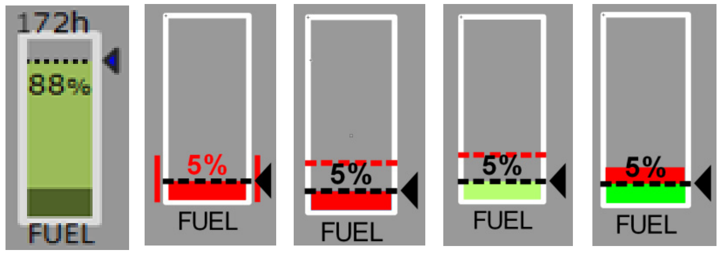

Figure 8: Examples of competing low fuel gauge designs. Note data displayed in this figure is fictitious.

2.3.1 Pitch-Roll-Rudder-Trim (PRRT) Gauge

The original gauge design for displaying ownship's pitch, roll, rudder, depth-in-bracket, and trim is depicted in (Figure 9). This graphic was very much overloaded in that a lot of metrics are displayed on the same space and reuse common axes. The original IID design describes the design of gauge in greater detail [1]; however, its functionality is summarized here for convenience:

- roll = rotation of straight black line about z-axis
- pitch = vertical position of black roll line along y-axis
- rudder = width of colored (e.g., green) rectangle
- trim = vertical position of colored rectangle along y-axis
- depth (in bracket) = vertical position of blue arrow along y-axis

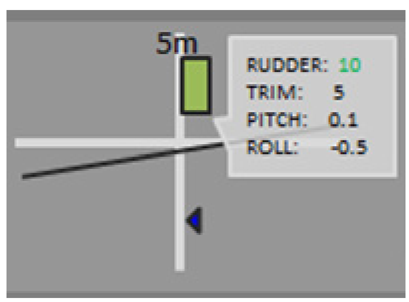

Figure 9: Original PRRT gauge.

Several issues arose during development and testing of this novel visualization. For example, none of the axes have labels or scale markers making it very difficult to discern values without purposely interacting with the graphic. The reference image used in the original design conveniently displays a state in which the various parameters are not overlapping; however, occlusion becomes a real problem when the moving constituents occupy the same space. The PRRT gauge includes a tool tip style feature for displaying digital readouts which exacerbates the occlusion issue. Perhaps more importantly, the visibility of rudder and trim values varies directly with the magnitude of rudder displacement. For conditions of minimal or no rudder displacement, the colored rectangle simply disappears with no way of representing trim values. Lastly, feedback from early usability testing suggested this visualization was simply confusing and difficult to interpret. In response to these accumulative issues the original design was scrapped entirely in favor of the three discrete gauges shown in Figure 10. The new gauges are discrete, single purpose, and leverage symbolism to represent ownship orientation. Each gauge is clearly labelled and values are represented both graphically by the symbol's orientation, as well as a numeric readout. All of the gauge's information is available at a glance; the user does not have to interact with the gauges to invoke details. Additionally, an up/down arrow indicates increasing/decreasing trends for each gauge's values. For consistency, each gauge uses the same 0 to ±30°scale for its analog component. However, during subsequent experiments it was observed that while rudder values can regularly displace to the maximum ±30°, ownship roll rarely exceeds a few degrees. Thus a smaller scale such as ±10° might provide a more pronounced indication of roll. Arguably the most significant drawback to this design is the amount of space required, especially in comparison with the original visualization. In this particular instance the depth-in-bracket and trim values were sacrificed entirely as the space constraints only allowed for the three (higher priority) gauges. The general lesson learned in this case seems to fall back to the "KISS" philosophy, in other words simple and discrete are preferable to complex and multipurpose. In any event, there is no substitute to usability testing for validating novel UI elements.

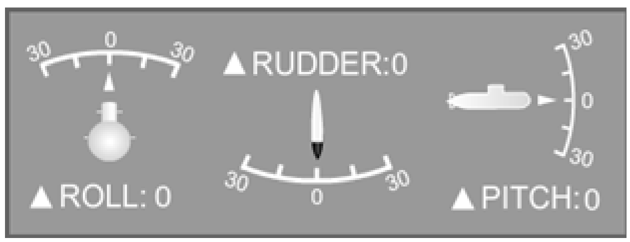

Figure 10: Redesigned pitch-rudder-roll gauge.

2.3.2 Telegraph

The last gauge with noteworthy changes in Area 2 is the telegraph used to display ownship's current speed setting. The original telegraph (Figure 11 left) lacked a gauge label and only displayed the current speed setting with no indication of direction nor adjacent values. The main challenge redesigning the telegraph relates to the horizontal and vertical space constraints. The Victoria-class has 10 discrete possible speed settings. Attempting to squeeze these values vertically within the telegraph gauge's allocated space results in font sizes that are far too small to be legible. The gauge's narrow aspect also does not accommodate full labels like labels "FLANK AHEAD" combined with discernible tic marks.

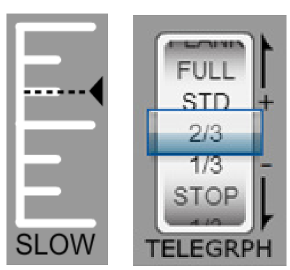

Figure 11: Original and modified telegraphs.

The solution eventually developed (Figure 11 right) uses a 'spin list' widget in which a portion of a vertical list (or wheel) of all possible values is displayed. The list automatically spins up or down such that the current setting is always centred vertically. Importantly, this design clearly indicates adjacent values as well as the current telegraph setting. The "AHEAD" and "REVERSE" prefixes were replaced by much more terse "+" and "-" indicators along the right side of the gauge. A potential disadvantage during testing with the spin list was occasional confusion regarding interaction (or lack thereof) with the spin list. Within the IID context, this gauge is for displaying output only, not for setting ownship speed (the IID is not an alternate helm). Note the spin list was adopted from a mobile component library. In general, mobile platforms and their UIs can provide a good source for inspiring spatially minimalist and touch friendly UI designs.

2.4 Area 3 – Ray Traces and Sound Velocity Profiles

Area 3 refers to the data visualization widget located at the bottom left corner of IID. It is used to display sound velocity profiles and ray traces. Only slight usability modifications were applied to this component. A pair of zoom buttons was added for adjusting the horizontal axis (range scale), the ownship symbols were replaced with more precise arrow indicators, and a reset button was also added that reverts to the default plotting parameters.

2.5 Areas 4 and 5 – Map and Contact Management

The overall tactical map itself (Area 4) saw very little changes from the original design; however, a couple noteworthy alterations were made to the map's accompanying menu (Figure 12). The menu layout was first updated to reflect its functional hierarchy; essentially map layers and filters were functionally organized and nested by either ownship or contact category. The menu category buttons (headers) were widened slightly and a small collapse/expand arrow was added left of the label. Clicking the menu headers toggles between the collapsed and expanded submenu states using a vertical slide up/down animation. This type of menu is also known as an accordion menu and is popular in smaller interfaces such as smartphones and tablets.

Arguably, the accordion menu can result in some ambiguity when all submenus of any particular menu are in the same state. More specifically, whether or not the header menu button should match the appearance of its children whenever they are all on or off. This behavior is particularly important when the menus are in the collapsed state, as the individual submenu states are not explicitly shown. The problem is further complicated when the sub menu buttons are in a mixed state (e.g., some ON and others OFF). The notion of a third toggle state for the menu header to indicate mixed sub menu conditions was briefly considered; however, its merits were not immediately obvious. Unfortunately, due to time constraints and higher priority tasks, this UI design issue went unresolved. As a result, the simple collapse/expand behaviour mentioned previously was (somewhat arbitrarily) chosen for the IID. Importantly, the menu header buttons do not indicate the collective state of their sub menu buttons.

The last change to the menus was the addition of touch-friendly vertical scroll (arrow) buttons that dynamically spawn as needed. These were added to help overcome the height constraints of the menu's allocated space.

Similar minor changes to the menu were also made in Area 5. Also in Area 5, the appearance of the lower tab navigation buttons (Figure 3) was altered slightly to better resemble tabs (vice toggle buttons). Although not implemented, a more consistent design might be to relocate the tabs to the top or side location (where tabs are more commonly found). Even more consistent design would have the selected tab's fill color match the color of the content panel's frame (e.g., the tab panel in Area 8 of Figure 1), instead of defaulting to blue.

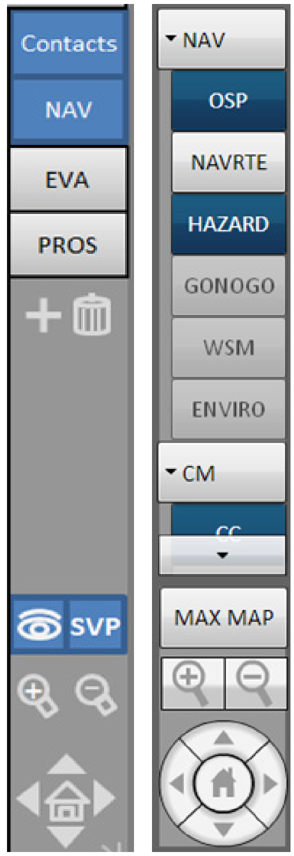

Figure 12: Original (left) and updated (right) Area 4 Map Menu.

2.6 Area 7 – Alert Panel

Inside the alert panel (Figure 13) the "Tactical" category of alerts was replaced with more specific categories; namely "Range" and "Nav". Within these new categories specific alerts were defined and labelled accordingly, thereby removing a lot of the ambiguity in the original design. An initial tap interaction enabled the user to click on an alert and display a popup window with details specific to the selected alert. However, early SME testing revealed a strong dislike to the occlusion caused by the popup window. As a result this interaction was disabled; however, an alert log was added to the tote panel in Area 8 in which the alert details could be viewed as required.

The alert panel's scrolling marquee (ticker) was resized to fit within the map of Area 4. The marquee's original design meant it would persist even when no alerts were presently active. To better utilize screen real estate, the behavior of the scrolling alert ticker was modified such that it was only visible when there was an active alert. Although the notion of text continuous scrolling across the GUI may seem fine in theory, in practice it had a tendency to become distracting and potentially annoying. As a result, interactions were added to allow the operator to toggle the scrolling and visibility of the marquee at any time.

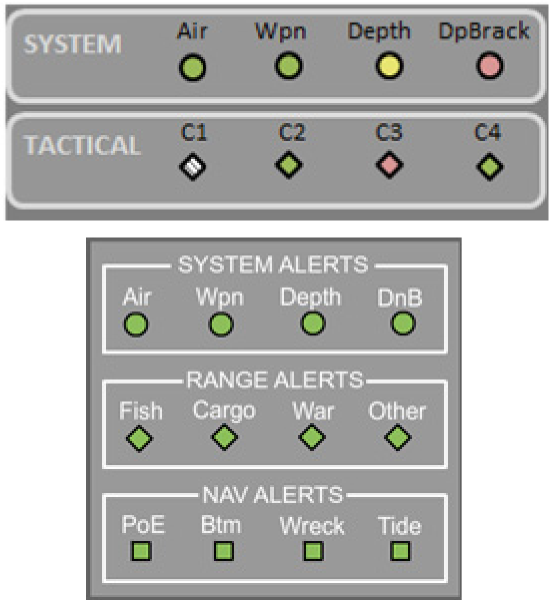

Figure 13: Original (upper) and redesigned (lower) alert panels.

2.7 Area 8 Dynamic (User Defined) Content

Most of the changes to Area 8 (Figure 14) were related to its structural layout and interactivity. As mentioned, both the visual design and interactive behavior of the main toggle bar for adding/removing content panels was redesigned for consistency and to better utilize screen real estate. Touch-friendly scroll buttons were also added to the toggle menu that spawn as required to enable vertical scrolling. Lastly, the weather and watch-list content panels were relocated to fixed persistent locations along the left side of the IID as per SME recommendations.

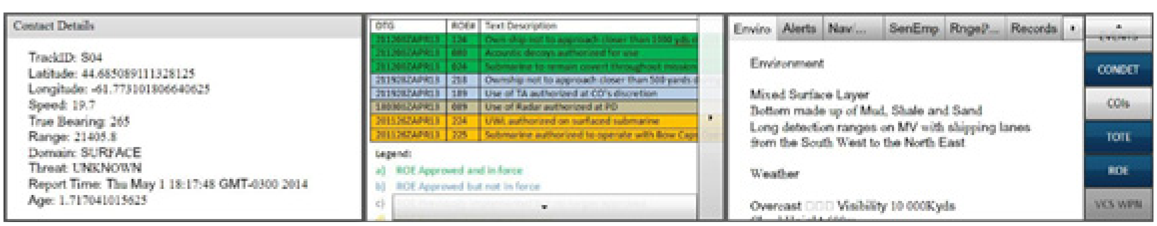

Figure 14: Updated Area 8 showing contact details, ROE's, and tote panels.

2.8 Other UI Design Considerations

2.8.1 Touch-Based Scrolling

Unlike existing operator stations in the submarine, the IID was designed with touch intended as the primary method of interaction. This choice was determined in part by the general use case for the IID; a shared situational awareness tool that requires relatively minimal interaction. Additionally, none of the proposed mounting locations were dedicated (e.g., seated) operator stations, but rather common (e.g., standing) areas within the ops room. The original design mapped the left, right, and double mouse clicks to single-tap, double-tap, and hover[2] touch interactions respectively. Notably absent from these interactions is a touch based equivalent for the mouse wheel (e.g., vertical scrolling). Lack of a mouse wheel became particularly evident whenever zooming and scrolling functions were desired. For example, whenever the size of content exceeded the boundaries of its container.

To enable scrolling via touch interaction, three different methods were explored. The first and perhaps most familiar technique was to spawn scrollbars as required. Unfortunately, the lack of precision inherent with touch interactions make traditional scrollbars awkward to use. Increasing the width of the scrollbars can alleviate the problem; however, the larger scrollbars occupy increasing portions of precious GUI real estate. Another scrolling strategy was to simply enable dragging of content within its parent container. This drag-to-scroll behavior is common on mobile GUIs as well as many map interfaces (e.g., for panning). One of the drawbacks to dragging content is the amount of relative scroll distance achieved with each drag gesture may be too small. Also, if the underlying content is not fixed, drag-to-select interactions (e.g., selecting text) are lost. Drag interactions can also become confused with other touch gestures. For example, attempting to scroll a menu by dragging may result in accidentally activating a menu button.

The third approach tested was to employ touch friendly scroll buttons with arrow labels to indicate direction (e.g., Figure 2). The scroll buttons can be operated in increment (single tap), or continuous (tap and hold) modes. Additionally, the scroll arrows provide a salient indication that more content (beyond the current view limits) is available and in which direction. Similar to the scrollbars, the space consumed by the scroll buttons is one of their more noticeable disadvantages.

The merits of each scrolling technique varied depending on the specific use case. Relevant factors include the horizontal and vertical space constraints, whether the scrollable content is static (e.g., an image), interactive (e.g., a menu list), selectable (e.g., text), and the sheer size of the content (e.g., large vs small map panning). Based on informal testing and SME input, scroll buttons were chosen as the best option to satisfy these criteria within the IID.

2.8.2 Symbol Overloading

The weather widget (bottom right Figure 1) uses repeated instances of a 'sun' icon to (in part) represent environment conditions like visibility, sun bearing/inclination, and sunrise/sunset times. Note within the IID, only the **visibility** instance dynamically changed, and all other sun instances

[2] The hover interaction was eventually interpreted as a touch and hold gesture.

were static graphics. This symbolism is fine during daytime sunny conditions, but if conditions are for example overcast, the scheme can (arguably) become confusing. In the cloudy state, if a sun symbol is still used to represent the sun bearing, then it could be misinterpreted to contradict the cloudy visibility icon (a cloud). This potential ambiguity can extend to other areas of the IID, for example the weather icon used in the original Area 8 toggle bar (Figure 4, second from right). This latter case also invokes a more general debate as to whether or not a button's icon should be a static graphic or dynamically updated to reflect current conditions. Unfortunately due to time constraints and higher priority tasks, this issue remained unresolved.

2.8.3 Runtime Customizations

Part of the scope of the original IID design included an interaction enabling end users to create and destroy custom menu options using "+" and trash symbols respectively (Figure 12 left). However, this advanced feature was abandoned early in the build process due to both the level of effort required and the absence of details in the design. Furthermore, there was debate as to whether customization was in fact desirable. The changes possible from one UI instance to another can present real obstacles both in terms of experimental repeatability and, more generally, interoperable usability. As example, if one user makes several significant changes to the UI, this could result in a less familiar or inconsistent experience for the next user. As well, the varying interface instances can have direct impact on training. For example, the steps required to complete a given task will vary depending on the custom alterations made to the UI. Ultimately, end user customization capabilities were minimal in the IID's instance, reserved mainly for choosing content to display in Area 8. Nonetheless, the dilemmas introduced with UI customizations generally remains an important consideration for future interface designs.

3 Conclusions

The IID concept contains a wide diversity of user interface elements, both familiar and novel. In the latter case, it is important to be aware that static GUI design methods can easily overlook issues specific to dynamic and interactive behaviors. This is why prototyping and testing are important iterative phases of a proper UI design process. The build and test phase(s) exposed several UI design issues within the IID. Both the breadth and rate of change of system states provided by runtime stimulation quickly revealed condition specific problems in various components of the IID. Similarly, interaction with functional prototypes exposed usability issues that might be difficult to discover otherwise. Generally speaking, mobile UI designs can be a good source of solutions for dealing with tight space constraints as well as issues related to touch input. Tried and true practices such as consistency and simplicity are still valid, while the pros and cons of more complex interactions, dynamic behaviours, and customizations can be less obvious and often require further evaluation on a case by case basis.

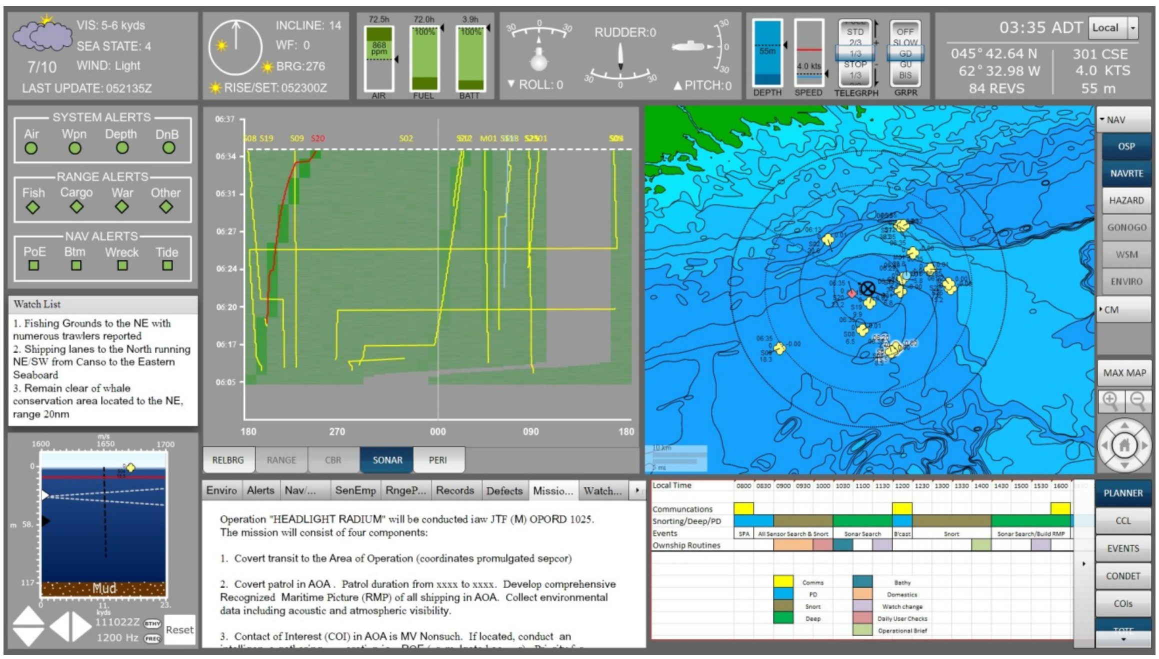

Figure 15: Redesigned version of the Integrated Data Display. Note data displayed in this figure is based on a fictitious scenario.

References

[1] B.A. Chalmers, "Developing an Information Integration Display for Submarine Command and Control", Proceedings of the Undersea Human Systems Integration Symposium, Vienna, VA, 2011.

[2] A. Gillis, "Using Games Technology for Maritime Research: a Case Study", presented to the NATO STO Modelling and Simulation Group (NMSG), Madrid, Spain, November 2013. https://www.sto.nato.int/publications/STO%20Educational%20Notes/STO-EN-MSG-115/EN-MSG-115-09.pdf.

This page intentionally left blank.

List of Symbols/Abbreviations/Acronyms/Initialisms

C2	Command and Control
DRDC	Defence Research and Development Canada
DTG	Date-Time Group
GUI	Graphical User Interface
IID	Integrated Information Display
MCTS	Maritime Command Team Support
OPS	Operations
PRRT	Pitch-Roll-Rudder-Trim
ROE	Rules of Engagement
SME	Subject Matter Expert
SVP	Sound Velocity Profile
UI	User Interface

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g., Centre sponsoring a contractor's report, or tasking agency, are entered in Section 8.)

DRDC – Atlantic Research Centre
Defence Research and Development Canada
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7
Canada

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

UNCLASSIFIED

2b. CONTROLLED GOODS

(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC APRIL 2011

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

User Interface Design Changes to the Integrated Information Display Concept

4. AUTHORS (last name, followed by initials – ranks, titles, etc., not to be used)

Coady, D.K.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
October 2016	22	2

7. DESCRIPTIVE NOTES (The category of the document, e.g., technical report, technical note or memorandum. If appropriate, enter the type of report, e.g., interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Reference Document

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

DRDC – Atlantic Research Centre
Defence Research and Development Canada
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
01ja	

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC-RDDC-2016-D053	

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The Integrated Information Display (IID) is a situational awareness display concept proposed for the command teams of Victoria-class submarines. A functioning prototype of the IID was constructed and trialed as part of a larger Victoria-class simulation. This report describes several changes made to the original IID design as a result of the development and test phases, including the specific User Interface (UI) issues each modification addresses.

L'affichage intégré de l'information (AII) est une vitrine sur la connaissance de la situation présentée aux équipes de commandement des sous marins de la classe Victoria. Un prototype fonctionnel d'AII a été construit et mis à l'essai dans le cadre d'une simulation plus vaste se rapportant à la classe Victoria. Le présent rapport décrit plusieurs changements apportés à la conception initiale de l'AII à la suite des phases de l'élaboration et des essais, notamment chacune des modifications liées aux problèmes particuliers de l'interface utilisateur.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g., Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

integrated information display; user interface; GUI; submarine; Victoria-class; Flash; prototype; virtual victoria; simulation