

Project Acronym: ILIAD
Grant Agreement number: 101037643 (H2020-H2020-LC-GD-2020 – Innovation Action)
Project Full Title: INTEGRATED Digital Framework FOR Comprehensive MARITIME DATA AND INFORMATION SERVICES
Project Coordinator: Netcompany-Intrasoft



Funded by the Horizon
2020 Framework
Programme of the
European Union

DELIVERABLE

D6.1 – Existing System Analysis Report

Dissemination level:	PU -Public
Type of deliverable:	R -Report
Contractual date of delivery	30/11/2022
Deliverable Leader	BUas
Status - version, date	Final – v1.0, 30/11/2022
WP / Task responsible	WP6 / T6.1
Keywords:	User experience design, user interface design, geographical information systems

This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101037643. It is the property of the ILIAD consortium and shall not be distributed or reproduced without the formal approval of the ILIAD PEB. The content of this report reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

Executive Summary

The goals of this informational and analytical report are

- To summarise the various ILIAD partner base systems that possess front-end interfaces;
- To analyse the strong user experience (UX) and user interface (UI) elements these base systems possess;
- To learn lessons for the further development of UX / UI designs of digital twins of the ocean both within and beyond Work Package 6 of the ILIAD project.
- To suggest UX / UI opportunities for these base systems and the design and development of digital twins of the ocean more generally.

To that end, the report offers 13 lessons on user experience and interface design derived from the analysis of nine base systems brought into the ILIAD project by its partners. As each lesson has its own chapter, and the chapter title is the lesson itself, we do not summarize the lessons here as well.

The report additionally offers key identified user experience and interface design opportunities for these base systems.

These lessons and opportunities will be brought along as Work Package 6 as well as the ILIAD project more generally continue with their end-user experience and interface design and development efforts.

The appendices offer more information on the user experience and interface designs of each base system itself.

Deliverable Leader:	Harald Warmelink (BUas)
Contributors:	Marie Lhuissier, Harald Warmelink, Alaitz Zabala, Joan Masó, Jonny Goddard, Simon Keeble, Javiera Lopez
Reviewers:	MEEO, RAMANI
Approved by:	Charalampos Ipektsidis (INTRA)

Document History			
Version	Date	Contributor(s)	Description
0.1	14/07/2022	BUas	Draft, incomplete, for internal feedback
0.2-0.4	04/11/2022	UAB, BLIT, BUas	Draft, complete, for ILIAD feedback & approval
0.5	29/11/2022	BUas, reviewers	Minor adjustments throughout the sections
1.0	30/11/2022	INTRA	QA and creation of the final submitted version

Table of Contents

EXECUTIVE SUMMARY	2
TABLE OF CONTENTS.....	4
DEFINITIONS, ACRONYMS AND ABBREVIATIONS	5
1 INTRODUCTION.....	6
2 ABOUT UX & UI DESIGN IN GENERAL.....	7
2.1 WHAT IS USER EXPERIENCE (UX) DESIGN?.....	7
2.2 WHAT IS USER INTERFACE (UI) DESIGN?.....	7
2.3 UX & UI DESIGN – STEP-BY-STEP PROCESS	7
2.4 UX & UI OUTCOME	8
3 LESSONS FROM THE BASE SYSTEM ANALYSIS	10
3.1 CONSISTENCY. KEEP REPEATED CONTENT UNIFORM.	10
3.2 MINIMALIST DESIGN. PRIORITISE AND LIMIT THE AMOUNT OF INFORMATION AND FUNCTIONS PER SCREEN OR SECTION.	13
3.3 STREAMLINE THE USER JOURNEY. ORGANISE (INTER)ACTIONS INTO GROUPS.	16
3.4 STATUS VISIBILITY & FEEDBACK. KEEP THE USER UPDATED ON WHAT THE APPLICATION IS DOING AND PROVIDE REGULAR INFORMATIVE AND RELEVANT FEEDBACK.	17
3.5 HELP & DOCUMENTATION	18
3.6 EASE OF USE. CATER THE EXPERIENCE TO BOTH NOVICE- AND EXPERIENCED USERS.....	20
3.7 DESIGN FOR ACCESSIBILITY	21
3.8 ALLOW CUSTOMIZABILITY	23
3.9 SEARCH & EXPLORE FUNCTION	24
3.10 SHARING & COLLABORATION	25
3.11 DISTINGUISH AND COMMUNICATE THE DIFFERENCES BETWEEN HISTORICAL AND PREDICTED DATA	27
3.12 SCREEN SIZE. DESIGN FOR VARYING SCREEN SIZES.....	27
3.13 AVOID DATA DUPLICATION	30
4 IDENTIFIED UX/UI DESIGN OPPORTUNITIES.....	31
4.1 VISTOOLS ANALYTICS.....	31
4.2 MARINOMICA	31
4.3 MSP CHALLENGE SIMULATION PLATFORM.....	31
4.4 ADAM.....	32
4.5 OCEAN DATA PLATFORM	32
4.6 RAMANI	33
4.7 NEXTGEOSS.....	33
4.8 NIMMBUS	33
5 CONCLUSION.....	34
6 APPENDIX – PARTNER APPLICATIONS.....	35
REFERENCES	59

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
UI	User interface
UX	User experience
WP	Work Package
WPL	Work Package Leader

1 Introduction

The goals of this informational and analytical report are as follows:

- To summarise the various ILIAD partner base systems that possess front-end interfaces;
- To analyse the strong user experience (UX) and user interface (UI) elements these base systems possess;
- To learn lessons for the further development of UX / UI designs of digital twins of the ocean both within and beyond Work Package 6 of the ILIAD project.
- To suggest UX / UI opportunities for these base systems and the design and development of digital twins of the ocean more generally.

The work that led to this report was as follows:

- The task leader (Breda University of Applied Sciences, or BUAs for short) requested ILIAD partners to identify the systems they were bringing into ILIAD with a clear graphical end-user interface. This led to a selection of these nine base systems:
 - VISTools
 - Marinomica
 - MSP Challenge Simulation Platform
 - ADAM (Advanced Data Management Platform)
 - Ocean Data Platform
 - Ramani
 - NextGEOSS
 - NiMMbus
 - MiraMon Map Browser
- For each of these base systems, an online focus group meeting was organised during which the system in question was presented, demonstrated and discussed, both generally and specifically on its UX and UI designs. These meetings were recorded and made available to the ILIAD consortium.
- Based on these meetings and their own expertise in UX/UI design, T6.1 partners analysed each base system. We observed various recurring UX and UI design patterns across the systems and noted which were particularly favourable. This information can be found in the appendix.

The remainder of this report firstly covers what the authors understand as UX/UI design. Subsequently, it dives straight into a set of important lessons in regard to the development of strong UX and UI design elements. Each lesson is accompanied by at least one example of an ILIAD partner application that scores strongly on it. In-depth content analysis, examples and suggestions are also offered. Not every applicable application will be mentioned under each lesson as this would result in a fair amount of repetition. The mention of a specific application over another does not necessarily mean it scores better, only that it proved to be an adequate example and no additional elaboration or example was needed to make a point. Finally, the report describes key further UX/UI design opportunities the authors identified for practically all applications. These are opportunities that could but might not be addressed during the ILIAD project. This report does *not* set an agenda for the remainder of the ILIAD project. That is simply beyond the scope of this report.

2 About UX & UI Design in General

2.1 What is user experience (UX) design?

User experience (UX) design is a process with the objective to design products that provide accessible, efficient and relevant experiences to users. UX design is about designing the experience of product use, the way users interact with said product, and the functions within said product. The main focus is enabling seamless goal achievement for the users.

2.2 What is user interface (UI) design?

User interface (UI) design is a process with the objective to design a 'front-end' interface with clear audio-visual communication and pleasing aesthetics, to aid in the efficiency of use and achieve the objective(s) set during the UX design phase. In a conventional application sense (e.g. a web or native application) UI design thus refers to the audio-visual appearance - 'the look and feel' - of an app that will aid the function of a product. It focuses on visual factors such as the colour palette, fonts, imagery, buttons, other interactive elements etc, as well as sound or audio effects of any kind.

2.3 UX & UI design – Step-by-step process

1. **User research.** Define and research the target audience. Learn about the user's behaviour, needs and goals, to build a strong foundation for the design strategy and create an optimally adapted product. Oftentimes, user personas will be created. Define the main use case(s) based on the acquired information.
2. **Interaction design & information architecture.** Consider what actions users would want to take using a product and translate this to a set of core functionalities and features. Create a user flow for each action, aiming to make it completable in a minimal amount of steps. Also, consider the general flow of the experience. Such a 'customer journey' should focus on helping the users find and achieve what they are looking for through an intuitive process. Think about the organisation, structure and labelling of content; content priority, the allocation of space etc. Create wireframes and/or mock-ups to experiment with different compositions.
3. **Visual design.** Consider what aesthetic response ('vibe' or 'feel') the corporate style should evoke, how to draw the eye to the core functionalities and features, how to visually convey content priority, font type and size, colour palette, white space, animated cues etc. It is important to take limitations or restrictions such as colour blindness into account in this design phase.
4. **Prototyping.** Experiment with ideas by making simple and testable mock-ups in which the various design directions from the previous steps get implemented and tested. Eventually, one- or a combination of prototypes will lead to the final concept that will get further developed.
5. **User testing & usability evaluation.** Test prototypes with the target audience to create a consistent feedback loop. Results from these tests might lead to a return to step 3 or 4.

- 6. Implementation & production.** Translate the acquired knowledge from the user tests into design decisions and implement the changes. Consider how to improve on the issues that users frequently run into. Repeat the 'user testing & usability evaluation' step until a strong final product emerges.

2.4 UX & UI outcome

What is important to consider for good UX design?

The 'user experience honeycomb system' by Peter Morville¹ considers seven important elements:

- **'Is it useful?'** The product must fulfil a user's need. Is it helping the users achieve their goal in a logical and cohesive way? What are the feature requirements to achieve this?
- **'Is it usable?'** How quickly and efficiently can a user operate the product?
- **'Is it desirable?'** There is more to a product than whether it is usable and useful. The product should be interesting and aesthetically pleasing to elevate the product experience. This is an emotive element, focusing mainly on image and branding than functionality.
- **'Is it findable?'** Is the product easy to find and access? Is the internal navigation to find information or reach a certain goal streamlined?
- **'Is it accessible?'** Do all users from different backgrounds and with different abilities experience the product in the same way? Is it usable for everyone?
- **'Is it credible?'** Provide a consistent and coherent service that the users can trust to make a product credible. It will greatly contribute to a positive user experience if the users feel the product is safe and secure.
- **'Is it valuable?'** Does the product bring value to the user? No matter how usable and desirable a product is, if it lacks inherent value, it is unlikely the user will return.

What is important to consider for good UI design? Multiple of the following design principles are part of Jakob Nielsen's '10 Usability Heuristics'.²

- **Status visibility & informative feedback.** Ideally, users should always be informed of what the system is doing through appropriate feedback within a reasonable time. Provide feedback for all user (inter)actions (Example: password strength). Immediate feedback will acknowledge that the application has received an input, reduce the level of uncertainty, reinforce the sense of manipulation, and prevent a user from potentially making mistakes (Example: clicking the same button twice). Actions should be organised into groups with a beginning, middle and end. Informative feedback at the completion of a group of actions gives users the satisfaction of accomplishment and a sense of closure (Example: A confirmation page after completing an order).
- **Error prevention & tolerance of mistakes.** Prevent errors from occurring as much as possible. (Example: by implementing smart defaults, confirmation requirements, greyed out unavailable items, preventing typing alphabets in number fields etc.). Provide feedback and a way out when an error does occur. Communicate the cause

¹ Morville, Peter (2004, June 21). *Semantic Studios*. Retrieved July 13, 2022, from www.semanticstudios.com/user_experience_design

² Nielsen, Jakob (1994, April 24). *Nielsen Norman Group*. Retrieved July 13, 2022, from www.nngroup.com/articles/ten-usability-heuristics

and how it can be rectified in the comprehensive language (Example: by implementing methods of cancelling, such as an undo and redo function). In the context of the systems selected for ILIAD, error prevention must also take into account the use and the potential for errors in numerous third-party systems.

- **Help & documentation.** Even though it is better if a product can be used without documentation, it is still necessary to provide help in case of need. Such documentation doesn't need to be visible at all times but should be easy to retrieve. The informational contents should be easy to search, focused on the user's task, list concrete steps and not be too large to avoid information overload.
- **Clarity / real world and product connection.** A product's interface should align with concepts, terminology and language already familiar to the user. Do not reinvent the wheel. Instead, reuse conventions and patterns that have proven effective in the past and exist across other platforms. Familiarity cultivates comfort and trust. It gives the users a sense of control, thus enhancing the user experience of a product.
- **Reduce cognitive load.** To maximise usability, reduce the load on memory by relying on recognition instead of recall. The number of cues that can help memory retrieval are much fewer for recall than recognition, making recall more mentally intensive and error-prone. The user should not have to remember information located elsewhere in the application in order to understand and properly interact with the elements on the current screen. As mentioned before under 'help and documentation', instructions for use of the system should be easily retrievable whenever necessary.
- **Consistency.** All repeated content within an application should be uniform: it should have the same visuals and functionality. Consistency limits the number of ways content is represented, allowing users to get familiar with the interface more easily and faster. This applies to everything from patterns to terminology. It ensures that users do not require to learn new representations for each task / action. Consistency creates a sense of control, familiarity and reliability, thus reducing the cognitive load and improving the clarity of the product.
- **Minimalist design.** Good UI design is practical, not decorative. Less is more. Overly decorative designs create unnecessary distraction from the elements that are truly relevant to the user. Do avoid overwhelming the users, define a main focus point for each screen, clearly convey its purpose and the required (inter)actions to maximise visibility and clarity. Avoid unnecessary complexity by cutting down on the amount of irrelevant or rarely needed information.
- **Ease of use.** An interface should remain simple and easy to use for the average person. Generally, users should not need training in order to use a product. Accommodate both novice and advanced users, while still tailoring the experience.
- **Accessibility.** It is important not to assume that users are like you, or the people in your direct environment. This goes for technical knowledge and abilities, but also opinions and general point of view. We often fail to consider minorities or people that are in other ways far removed from ourselves. It is important to design with our possible differences in mind, such as motor, cognitive or visual impairments.

3 Lessons From the Base System Analysis

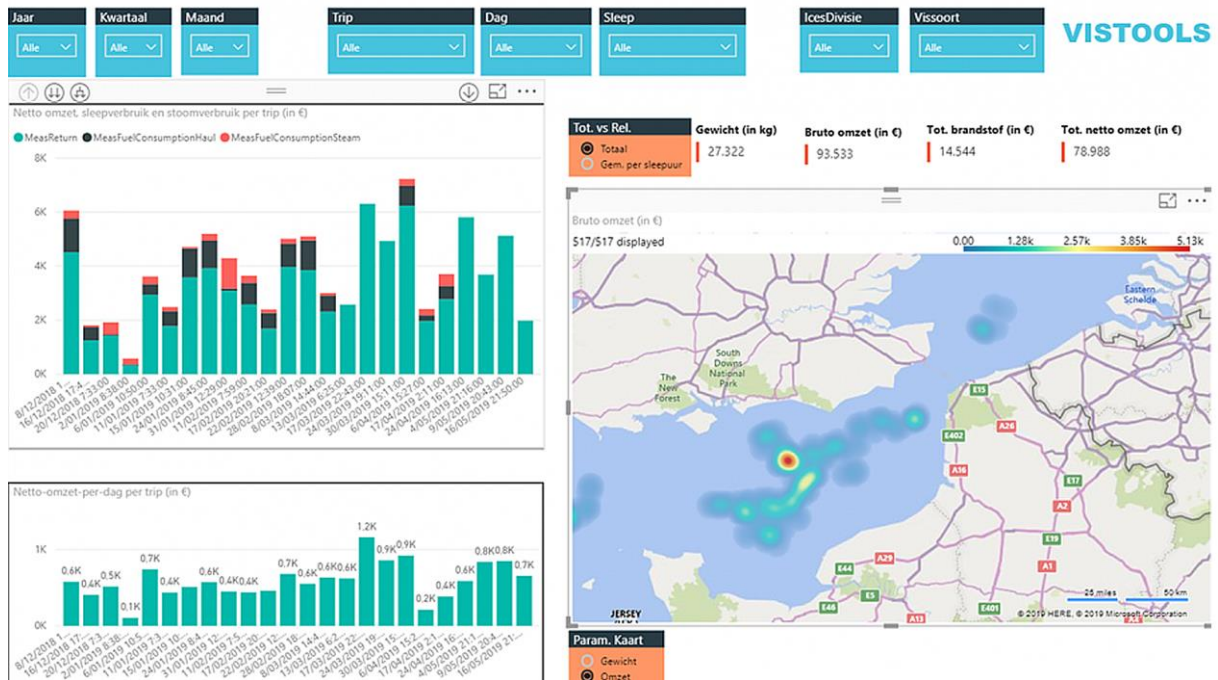
3.1 Consistency. Keep repeated content uniform.

Applications often have repeated elements across various screens, functions and features. It is important for these elements to be consistent. Regularly referring back to a style guide during the design process helps safeguard the overall corporate style of an application.

As mentioned before, consistency limits the number of ways content is represented, ensuring that users do not have to learn new representations for each task / action. Consistency creates a sense of control, familiarity and reliability, thus reducing the cognitive load and improving the clarity of the product.

- **Fonts.** Implementing a couple of font types and sizes helps indicate the content priority. It creates a natural division between different content (both visual and written) and indicates to the user in what order content should be ingested. Consider how to implement the various font types and sizes, such as headers, regular text, quotes, menus etc. Once the various font types and sizes have been assigned a certain function, they should be used consistently.
- **Images.** Consider what kind of content would benefit from supportive imagery and make sure it is provided consistently. Translate certain written content into- or combine it with visual representations, such as icons. Reducing the number of text elements frees up space, creating more breathing space and thus enhancing the clarity of the interface. Other benefits of the use of icons is that they break language barriers and are universally understood. Just like all the other user interface elements and building blocks, icons should have a coherent style (outline, glyph, coloured, flat, badge, two-tone, 3D etc.)
- **Colour palette.** Pick a minimalist colour palette with only a couple of accent colours. This helps draw the eye to important content and functions. Use the same accent colour for all the functions and features of the same type.
- **Colour Contrast.** Depending on a user's perception of colour, particularly those with colour blindness, some users may not be able to properly perceive all aspects of a design with poor colour contrast levels. Good colour contrast is a critical aspect of the Web Content Accessibility Guidelines (WCAG) and an accessible UI should employ 'soft-proofing' techniques to colour schemes.
- **Navigation.** Navigation elements such as menus, back- and continue buttons, close buttons, links (etc.) should look and behave similarly across the whole application. Applications that deal in knowledge/information sharing should effectively categorise excessive information to avoid confusing users.
- **Terminology.** The user interface should align with terminology and language familiar to the user, and has to remain consistent across the whole application. When a naming convention is introduced to describe a certain screen, function or feature, it should be implemented everywhere. Consistency in terminology doesn't only apply to the exact use of wording but also to the quality of the written content. Casual and corporate language shouldn't be used interchangeably. Failing to do so risks negatively impacting the credibility of the application.

Examples:



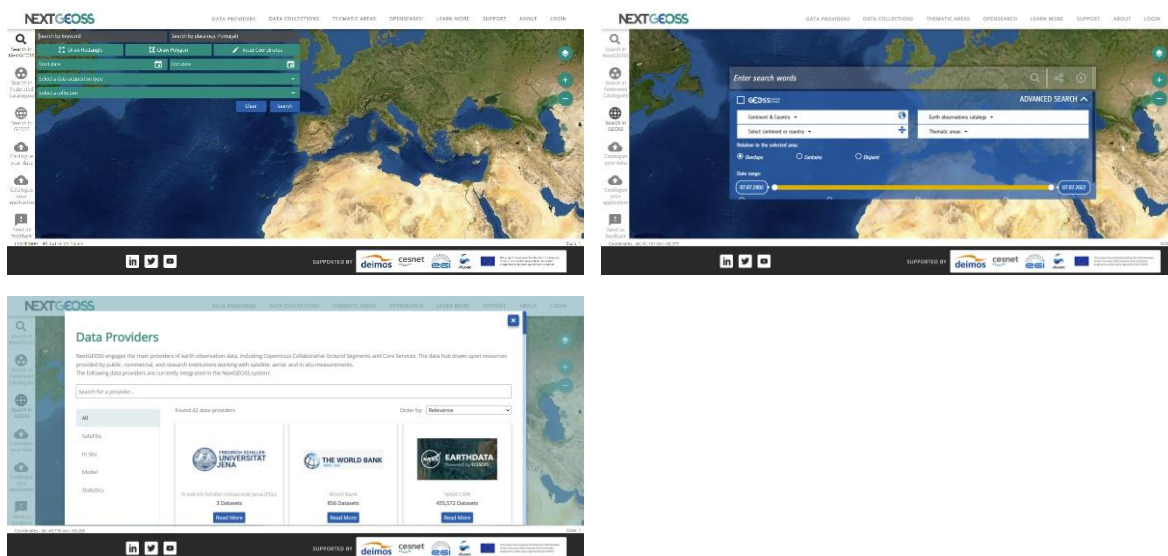
VISTools

VISTools has a strong base when it comes to the consistency of the user experience and user interface.

The application has a white background, while implementing strong accent colours to indicate important functions, features and interactive elements. All elements part of the same family (example: the blue search filters) have the same accent colour, clearly indicating that they belong together and share the same functionality. The amount of accent colours used in the application is slightly elevated, but as they are clearly restricted to specific areas within the screens, it doesn't feel too overwhelming.

The application clearly separates the various functions and features within the screen, using adequate breathing space. Wherever possible, information is conveyed visually, avoiding large areas of text. The interaction with all the different graphs is consistent. The drop down menus behave the same across the whole platform etc.

A few different tabs are available, each focusing on specific thematic information. This makes sure the content is to the point and relevant.



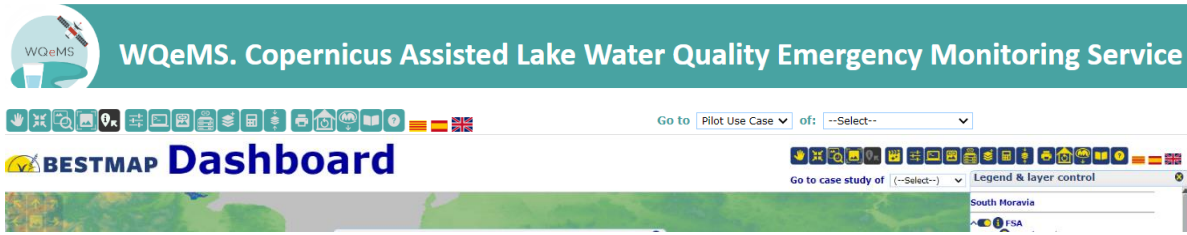
NextGEOSS

NextGEOSS has a few areas in which the consistency of the user experience and user interface can still be improved.

The application does limit the use of colour effectively to two accent colours, while keeping other elements white and anthracite grey. However, these accent colours get used interchangeably in what they represent. They do not always indicate the same kind of interactive possibilities or content type. For example, the majority of the time, blue is used for interactive buttons such as the close- and search buttons. However, on various pop-up screens, the 'add feedback' button is visualised in green.

Another example are the 'search fields'. Certain fields are green while others are anthracite, without there being a clear functional or thematic difference between the two.

When it comes to the user experience, the application contains six menu tabs to access data sets from various sources. While it is generally beneficial to have various search- and discovery methods, it doesn't make much sense to disperse features that are so similar in functionality. Separating these features over multiple tabs and two menu bars creates the risk of confusing and disorienting the user. They might fail to understand the difference between the various options. Despite the similarity in functionality, three types of visuals have been implemented across these features.



MiraMon Map Browser

Icons and buttons are in SVG Scalable Vector Graphics (SVG) format with two basic colours. The two colours can be personalized for each map browser instance in harmony with the colours of the project and its logo.

3.2 Minimalist Design. Prioritise and limit the amount of information and functions per screen or section.

It is valuable to define a clear target audience and carry out extensive user research to get to know the users. This will help determine what users want to do, what their main goal is and what they expect of a specific application. Creating a user flow, a visual representation of the journey taken by a theoretical user when using an application to complete a task, can be a very helpful method to design the information architecture.

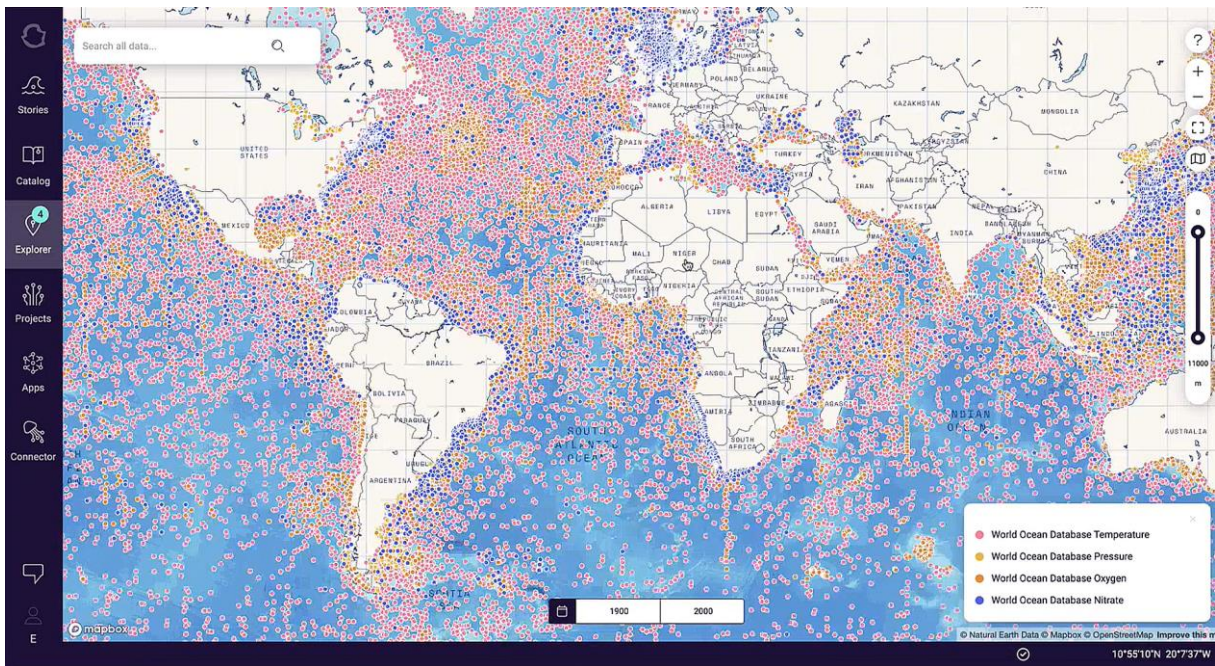
Good UI design is practical, not decorative. Overly decorative designs create unnecessary noise and distract from the elements that are truly relevant to the user.

To avoid overwhelming the users, define a main focus point for each screen or section, clearly convey its purpose and the required (inter)actions to maximise visibility and clarity. Avoid unnecessary complexity by cutting down on the amount of irrelevant or rarely needed information. Every element on every screen should be absolutely essential.

Make sure that the information architecture is simple, logical, and clearly signposted. Consider the allocation of available space. Important information, functions and features should be prioritised and placed at the forefront. This type of content must be easy and quick to find through a streamlined, flat hierarchy navigation. It should draw the most attention on individual screens, while less frequently needed information can be hidden out of sight. Do keep in mind that even less frequently needed information should still remain easily findable and accessible.

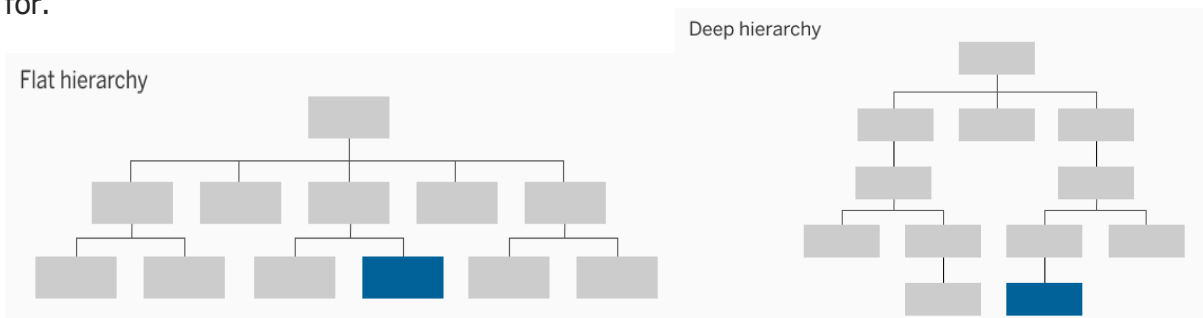
The visual elements to consider to achieve a minimalist design have previously been discussed in the consistency lesson.

Examples:



Ocean Data Platform

The Ocean Data Platform application has a very clear, to the point navigation system in place. There is a single menu bar located on the left side of the screen with 'flat navigation hierarchy', meaning there are a limited number of levels within the navigation. The fewer levels the users need to go through, the quicker and least confusing it is to reach the content they are looking for.

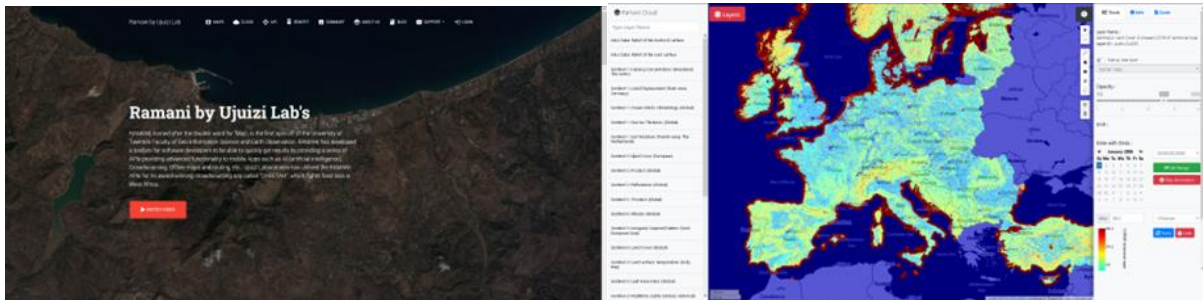


If the amount of content within an application allows for the use of a single menu bar, this can bring several benefits. In the case of Ocean Data Platform, it makes it clear how the user is supposed to navigate through the application in a single glance, as there are no additional options dispersed over various menu bars. It also gives the user a rough estimation of how many different (important) screens the application possesses. Lastly, through the use of clear naming conventions for the tabs and the support of visual icons, the application communicates to the users what kind of functions and features can be expected on each screen.

This minimalist design successfully extends into the various screens of the application. Each screen focuses on specific thematic content. The information within each screen is relevant and kept to a bare minimum. Certain screens, for example the 'catalogue', require a few

(inter)actions to make optimal use of its functions and features. Each of these steps is clearly defined, with an obvious separation between them. At no point does the application show an overwhelming amount of information all at once. Combined with adequate use of breathing space, this makes for a very calm and clear user interface.

One area where the Ocean Data Platform application could improve on its minimalist design would be the interactive map on the 'explorer' screen. Instead of visualising an enormous amount of separate dots, a gradual heat map could be implemented. (As seen for example in Marinomica, among others.)



Ramani

Ramani application has a simple and harmonic interface, with a single menu bar with eight options (some of them with drop-down options), some of which are still under development. The minimalist design of the cover shows a screen-size satellite image in dark colours, with a paragraph explaining its history and trajectory, written in white letters. The contrast between the two parts focuses the attention on the written paragraph, while the image is left behind as background.

Below are three screens with a simple layout (the back cover is blank and there are no images). Each screen focuses on different topics, with the respective items and their description.

Once logged in layers could be better organized per satellite groups within drop-down options. This would improve the search experience.

MiraMon Map Browser

This is a counter example of the minimalistic design. As a general tool, it needs to make available as many functionalities as possible. However, there is a config.json file that allows an administrator to only make visible the functionalities that are really needed by a certain user group for a selected purpose.

3.3 Streamline the User Journey. Organise (inter)actions into groups.

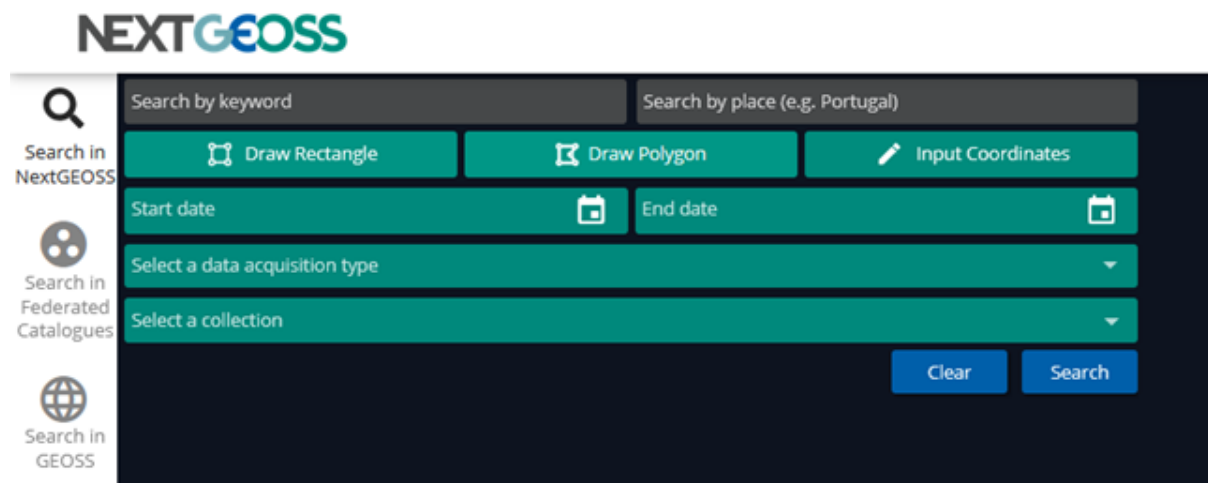
All functions and features in an application can be seen as a group of (inter)actions, a group of steps that need to be taken to achieve a specific result. It is important to define a clear start, middle and end for these groups. The completion of a group of actions gives users the satisfaction of accomplishment and a sense of closure.

Limit the number of actions a user has to take within a group to no more than five steps. This is to avoid the users 'getting lost' in the quantity of the content and information overload. Feedback should be provided at (almost) every step to assure visibility of status. Users need to know where they are in the application, where they are in a process and whether they are heading in the right direction when it comes to achieving their goal.

It is equally important to streamline the user journey and interaction within the application, as it is leading up to it. Everything should be clear and to the point, every step of the way. It should be clearly indicated how an application can be accessed. Where can the user find the application? Does the application require to be downloaded and installed?

The more steps are implemented, the higher the chances of losing the user along the way. This reduces the findability and, therefore, also the usability of the application. If the process is long and complex, there is a risk the user will back out because he gets confused or due to the lack of time. If a user is lost so early in the process, where they have not yet been able to experience and appreciate the value of an application, there is a chance they won't return at a later point as their first experience turned out unpleasant or undesirable.

Examples:



NextGEOSS

The search area of the platform has different boxes to detail the data search, and once a parameter is set, it remains visible in each available area. This resource could be exploited and in order to improve the user experience as they search, the boxes could change colour once that the information request has been fulfilled.

3.4 Status Visibility & Feedback. Keep the user updated on what the application is doing and provide regular informative and relevant feedback.

In the previous lesson, we discuss how an application consists of multiple functions and features that require various amounts of (inter)actions to be optimally used. These (inter)actions are steps that need to be taken to achieve a certain goal.

An application should provide feedback to the user at (almost) every step they take, to assure visibility of status. It is important to make objects, actions and options visible. The distributed nature, including the use of third-party systems in some of the systems selected for ILIAD, make this design aspect particularly challenging and the visibility of external actions is not always going to be possible.

The user needs to know where they are physically located in the application (what screen) and where / how far they are in a specific process (for example when using a specific function or feature). This is important so that the users know if they are going into the right direction in order to achieve the specific thing they set out to do. This also reduces the load on memory and recall, making an application more usable.

A few examples of when to implement feedback elements to achieve adequate visibility of status:

- Provide cues when selecting a user interface element. Implement a little animation, a change in button colour, a change in button or link colour on 'mouse-hover', sound cues etc.
- Implement a required confirmation request everywhere where a user can apply major / serious changes. Apply error prevention and be tolerant of mistakes, for example by providing undo or redo options wherever possible.
- Provide warning messages with helpful hints, for example in places where the application misses certain required information to properly function, such as an incomplete form.
- Provide confirmation messages after the completion of a function or feature. For example, a thank you page with an overview after placing an order or a confirmation message once a form or message has been sent.
- Provide error messages and explain what the user can do to rectify it whenever they cause- or the system runs into issues.
- When the system requires more than a few seconds to load it should give the user feedback through either infinite loading indicators (typically used for a short waiting times) or a progress bar (typically used for waiting times longer than 10 seconds). For example, if a large database is being retrieved from an external source.
- Do not let users wonder if the system crashed. As mentioned above, show how far it is in the process and give a warning if the system times out and the content can't get (fully) loaded.
- Specific to data visualisation applications: Send the user to the right place on the map when a very small / localised data set is activated, instead of having them try to figure out where or if the changes have been implemented on the map.

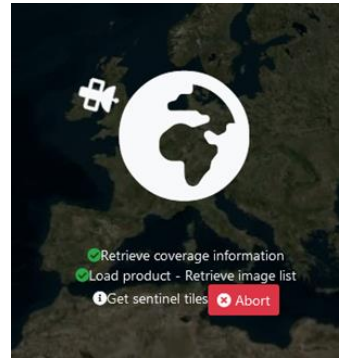
- If an error has been encountered, the user should be provided with a clear means to report the error. A user is much more likely to return if they feel that issues are likely to be resolved in a timely manner.

Examples:

Marinomica



ADAM



Marinomica & ADAM

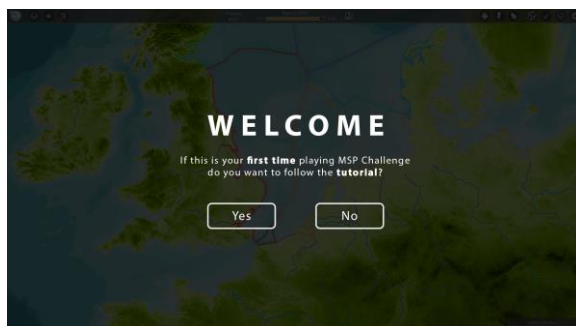
Both ADAM and Marinomica clearly communicate to the user when content is being retrieved and they are required to wait for a layer or process to load.

An infinite loading indicator like the one used in Marinomica is perfectly adequate, especially if the waiting time is limited. ADAM however, takes it a step further by providing the user with intermediate updates, specifying which steps have already been successfully fulfilled and which are still pending before the full completion of the task. This gives the users a bit more insight into what the application is doing and how long the remaining waiting time is likely to be.

3.5 Help & Documentation

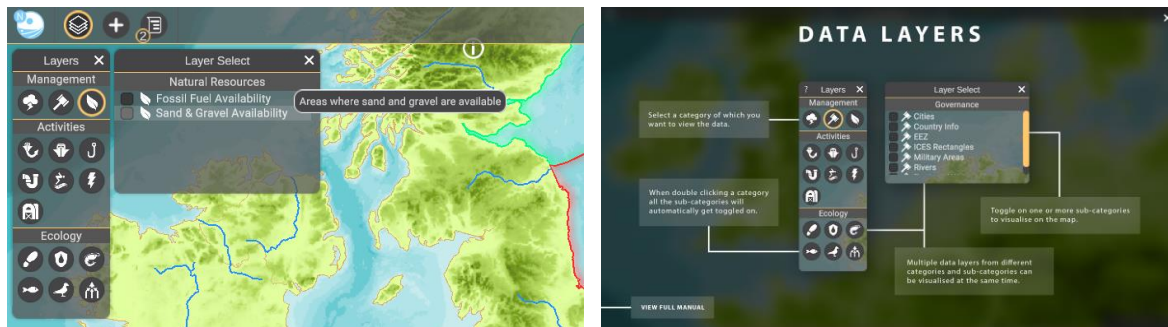
Ideally, an application should be usable without documentation. However, instructions should always be easily accessible in case of need. This can be minor documentation and hints, provided to the user through 'on hover' information or a help option. Another way to achieve this could be to give the users the ability to turn on a step-by-step tutorial, consult a knowledge base wiki, or contact the helpdesk.

Examples:



MSP Challenge

A step-by-step tutorial has been designed and will soon be implemented. New users will be prompted to follow the tutorial when first starting a new session. The tutorial will be retrievable through the settings menu, even after closing or completion, if the users so desire.



Almost all elements in MSP Challenge, such as buttons and windows, have 'on hover' information and hints. Soon, a help (?) button will be added to every window that will provide the users with an informative and visual overview of a window's functions.

MSP Challenge also has a knowledge base wiki. This website can be accessed through the application while playing, for example, to consult information about various data sets. It can also be used standalone, outside of the application if the user for example requires information on how to instal the software.

thanks for joining the ADAM users community.

In order to optimize your experience on eodataservice, we have prepared a 4-minutes video[1] to describe the platform concept; we invite you to have a look at it.

Moreover, to provide some usage hints, we have set-up a YouTube channel[2] with short videos focused on specific applications, such as:

- study of agricultural parcels using landsat 8, modis and NOAA precipitation data (3:28) [3]
- burnt areas detection using Sentinel 2 data (4:16) [4]
- computation of statistical parameters using processing functions (2:46) [5]

We will keep you posted once new datasets and new functionalities will be made available.

Feel free to provide your feedback and notify any issue at contactus@adamplatform.eu.

1. <https://www.youtube.com/watch?v=YUmhalnOMsQ&feature=youtu.be>
2. <https://www.youtube.com/channel/UCqy-bd6CpXMOlRqfHmBqZw>
3. <https://www.youtube.com/watch?v=RG5bl-Un0VM>
4. <https://www.youtube.com/watch?v=X2M68GJHudk>
5. <https://www.youtube.com/watch?v=TmmwqJc408>

This message was sent by the [ADAM platform](#).

ADAM

ADAM is another example of a base systems that provide the users with helpful 'on-hover' information. Additionally, once the user creates an account, they also receive an automatic welcome email with different data links to videos that explain the general operation of the site and provide tutorials of specific applications, with the purpose of facilitating the user experience.

SatCat 2.0: Catalan Data Cube

UTM31N - WGS84



MiraMon Map Browser

This app has a generic help page accessible through the help button. In the future, it could improve the user experience with more focused and contextual information.

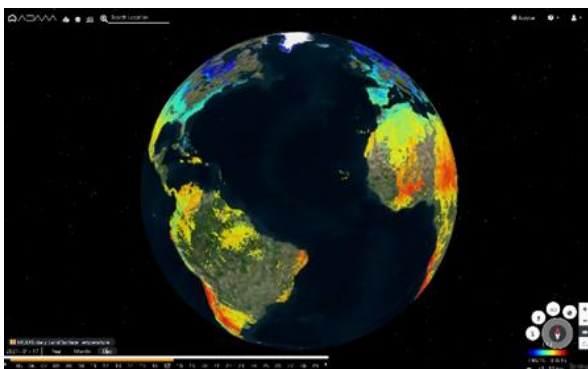
3.6 Ease of Use. Cater the experience to both novice- and experienced users.

An application should be simple and easy enough to be used by an average person, while also catering the experience for both novice- and experienced users. By clearly distinguishing the indispensable- and the optional, more in-depth content, you allow users to pick more adapted path to their experience level.

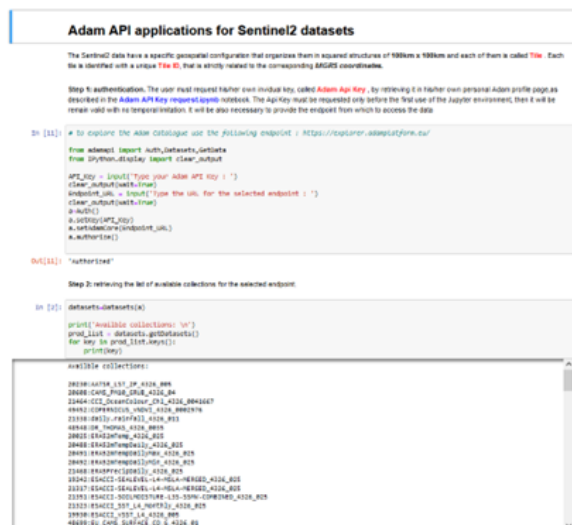
First time- and novice users can follow a more basic, simplified path while more experienced users can decide to use accelerators, shortcuts and access the more in-depth content.

Examples:

Online user version



Developer version



Ramani, Ocean Data Platform & ADAM

Ramani, Ocean Data Platform and ADAM all implement various interfaces. The front-end interface is visually intuitive and usable by an average user, while the back-end Jupyter Notebook interface provides more in-depth features for experienced users, such as scientists.

In the case of ADAM, one of its objectives is to “remove the main barriers to realize the democratization data process”. ADAM is designed to be used by various types of users, with different levels of experience and technical knowledge.

The “explorer” interface (shown on the left) is targeted to users who prefer a comfortable web GIS interface, while the other interface (shown on the right) is targeted to users with additional data processing needs, through Jupyter Notebooks, where they can input live code with detailed documentation.

3.7 Design for Accessibility

The design of an application often mainly revolves around one target audience. User personas get created based on the average values of users within a target audience. Though this is considered a useful- and generally effective method to determine the general wants and needs of the users, the outcomes can sometimes be a bit stereotypical and superficial.

We often fail to consider minorities, people with functional diversity, who might still make up a significant percentage of our target audience. Think of users with motor, cognitive or visual impairments (e.g. colour blindness.) This is why it is important to design the user experience and user interface with accessibility in mind.

The following elements can be considered to make a design more inclusive:

- **Navigation.** Allow multiple ways of navigation within an application for users with motor impairments. For example: keyboard only navigation, supporting gestures, eye tracking etc.
- **Visual cues.** Do not rely on a single type of visual cues to convey important information. For example, combine colour and pattern for users with colour blindness; implement motion animation that draws attention for users that have bad eyesight.
- **Non-visual cues.** Implement non-visual cues to convey important information, for users with visual impairments. For example, vibration in a controller or mobile phone, warning / error sounds, key-press audio, a text to speech function etc.)
- **Customizable colour profiles.** Allow users to change the colour settings so they can create combinations that are more strongly distinguishable in case the defaults are not adapted, for example for users with different types of colour blindness.
- **Customizable brightness and contrast.**
- **Customizable user interface- / text size.** Allow users to change the scale of the text and UI to suit the users' preference and/or requirements.
- **Customizable language.** Accessibility is not only about making applications accessible to users with medical impairments. It is about making it inclusive for users with any kind of difference in background, including cultural. Translating an application in at least a few common languages allows the user to pick an option that they feel most comfortable with. To take this a step further, both ‘left to right’ and ‘right to left’

reading can be implemented, the latter being more common in Japan and Arab countries.

Examples:



NiMMbus

This app supports three different languages: English, Spanish and Catalan at the time of writing. Users can set the UI language as preferred both before and after logging in. Once configured, the server will remember the choice and it will display the user interface in the selected language every time the user logs in.

Marinomica

Once a layer from the catalogue has been selected and loaded into the platform, the legend (in the bottom-right corner) indicates the range of graduated values with the colour's palette. Upon clicking the palette, customization options appear offering ten colour palettes to choose from to fit the user's visualization needs or preferences.

The customization options include two sliders. The first is to adjust the minimum and maximum values, the second to select the layer's opacity percentage. Both can also be used by entering specific values. As the sliders are adjusted, the map display instantly adjusts accordingly, allowing the user to more quickly and easily determine the proper values.



MSP Challenge

In MSP Challenge, the user interface size can be adapted to the user's needs. This accommodates users with visual impairments and allows them to still optimally use the application.

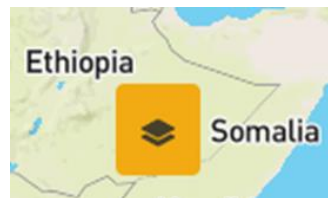
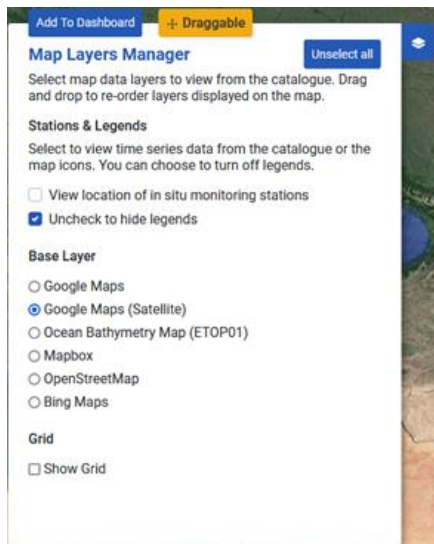
3.8 Allow Customizability

As mentioned in the previous lesson, allowing visual customizability enhances the accessibility, but it also makes an application more desirable.

Not only does it support users with functional diversity, but it also allows average users to simply personalise the user interface in a way they deem aesthetically pleasing and organised. A company could also customise visuals adapted to their use case or company branding. This could be considered to be mostly 'eye candy', but its effect on desirability should not be underestimated.

In the case of data visualisation applications such as many ILIAD partner applications, allowing customizability can also make for a stronger analysing tool. Customising visual elements such as the colour palette or transparency allows users to better understand and compare the data.

Examples:



Marinomica

Marinomica has different options to personalize the experience of the user. First,

once logged in there is a panel where the user can choose different base maps, all of them from well-known software (Google Maps, OpenStreetMap, Bing Maps, etc). Furthermore, there is an option to activate or deactivate a Grid showing by default parallels and meridians.

This panel can also be dragged and placed wherever the user prefers on their screen. Simultaneously, the panel is collapsible or expandable. Once expanded the button turns blue, when selected it changes to yellow.

MiraMon Map Browser

A config.json file allows an administrator to configure several interface aspects and personalize the browser in terms of object positioning, colours, hiding functionalities, data layers, presentation styles, etc.

ADAM

Once the user has loaded a product, they have the possibility to adjust the opacity percentage, change the color table, and choose the equalization method.

3.9 Search & Explore Function

Users usually have a general idea of what they can do and find by the time they have decided to use a specific application, but they aren't necessarily aware of every single available option and function. Think of all the types of data sets available in the various ILIAD partner applications, all the clothes in a shopping app or the type of courses on a university website. We know what the platform is for, we know roughly what type of information and content to expect, but not what the information, functions or features actually contain.

A part of the users will already know what they want and need. The application should allow them to easily find that content and achieve their goal(s). A streamlined navigation system, clear menus, consistent naming conventions and labelling, and even something as simple as the implementation of a search bar, are all elements that can make the process of finding what they are looking for easier and quicker.

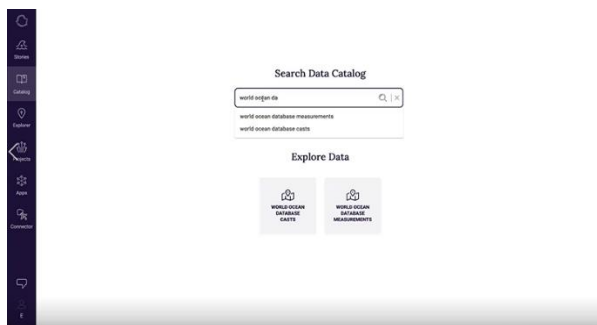
However, there will always be a group of users that do not know exactly what they are looking for. They might be on the lookout to be inspired, but they might also simply not be aware of what is possible or exists, what content is already out there etc. This part of the audience has no need for a 'search' method, it needs a 'browse' method. The application should allow for them to explore content. This can be done by simply browsing a full list of content, but even better is to provide the users with clear categories through which they can filter or view thematic applications or suggested content.

A few examples of explore functions:

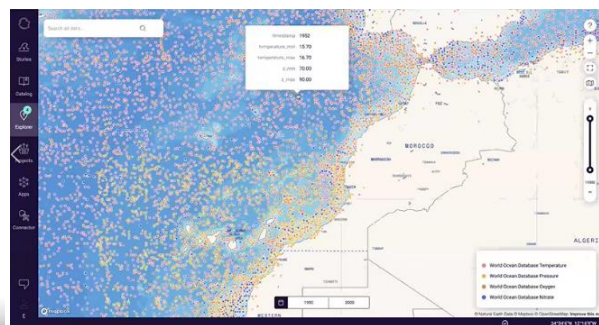
- Various ILIAD partner applications allow to search for data sets based on popular categories or thematic applications.
- H&M recommends full outfits for the upcoming seasons. This gives the users inspiration and particularly helps less fashionable / fashion driven users.
- Deliveroo and Uber Eats shows the restaurants and pubs in the user's area and also allows to filter per type of cuisine (country, vegetarian, vegan etc.)
- IKEA sharing their customer's home setup photos and tagging the represented products on their website and app. This gives the users inspiration and provides them with some insight on how to use and combine certain furniture and objects.

Examples:

Catalogue area



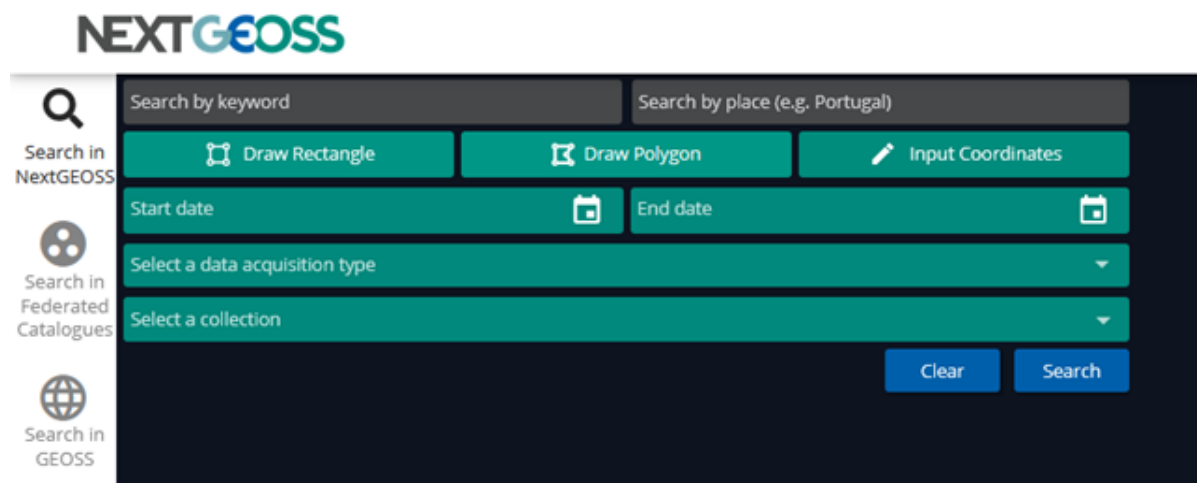
Explore area



Ocean Data Platform

In this application, the search function is provided on the 'Catalog' page (which can be accessed through the menu bar on the left-hand side). Once the users have input their keywords of choice, a filtering option appears to help refine the search.

To stimulate exploration, mainly for users that do not yet know what it is exactly that they are looking for, four layers with global scale data points are displayed by default on the main map. It allows one to zoom into the areas of interest and review the available data, as well as visualizing polygon data to facilitate the understanding of the phenomena in question.



NextGEOSS

The search area of this application is divided into multiple sections to facilitate more detailed and pinpointed searching, but none of the options are mandatory to fill in.

The sections in the first row (anthracite) allow the users to search by concept (such as keywords or locations). The sections in the second row allow to search a specific polygon area that the users can select / draw themselves on the map (fixed rectangle, free drawing polygon or by coordinates). The sections in the third row allow to search by date. The section in the fourth row allows to search by data type and finally, the section in the fifth row allows to filter the search results by predetermined data collections.

3.10 Sharing & Collaboration

Specific to the ILIAD use case, if the overarching goal of an application is to help improve communication, collaboration and planning around ocean spatial planning in the future, it is worth considering creating sharing and collaborative options instead of single user applications.

- Consider models that support multiple users to simultaneously work and collaborate in an application.
- Allow users to easily share and / or merge the work they did individually with that of other users.

- Allow embedment of content, such as dashboards, into other applications, websites or presentations.

Examples:



PA general datasets

- Protected Area Location
- Site type
 - International Marine Protected Area
 - Large Marine Ecosystem (LME)
 - National park
 - Natura 2000 site
 - World heritage
 - Biosphere reserve
- DEIMS-SDR
 - Nationally accredited LTER site
 - ECOPotential sites in DEIMS

Swiss National Park datasets

- Swiss National Park Limits
- Digital Terrain Model
- Bare Ground (~2010)
- Vegetation Metrics
 - 2012
 - Above Ground Biomass (AGB)
 - Canopy Height Model (CHM)
- Land use/cover
- NDVI Landsat
 - 09-04-2016
- NDWI Landsat (Normalized Dif)
 - 09-04-2016
- Sentinel 2 L2A

Feedback

Add user feedback to the layer "Vegetation Metrics, Canopy Height Model (CHM)"

[Add a user feedback](#)

Previous user feedback to the layer "Vegetation Metrics, Canopy Height Model (CHM)"

JoanMaso, 2022-07-07 09:13:22
 NiMMbus Id.: 75735B439U6WEWN265Q13YJ215E475848K1H056VGQ1KF40
 Abstract: Test
 User role: General public
 Date (creation): 2022-07-07
 Rating: 5/5
 Comment: I like the previous comment and its honesty

Target resource (Primary): Swiss National Park Vegetation Metrics, Canopy Height Model (CHM)
 Identifier: http://maps.ecopotential-project.eu/cgi-bin/MiraMon.cgi/SwissNationalParkVegetationMetrics_VALUE{DIM_METRIC_CHM}
 NiMMbus Id.: 1DF54MMTK0P7TFU0979J5UTR3S670U08150DP5QCOH083YI

AlaitzZabala, 2022-03-31 15:08:46
 NiMMbus Id.: 15467A1W3W8H1227RX4961968E5JQ0XA6YLM8MW71KR290K
 Abstract: New version to be released
 User role: General public
 Date (creation): 2022-03-31
 Target resource (Primary): Swiss National Park Vegetation Metrics, Canopy Height Model (CHM)
 Identifier: http://maps.ecopotential-project.eu/cgi-bin/MiraMon.cgi/SwissNationalParkVegetationMetrics_VALUE{DIM_METRIC_CHM}
 NiMMbus Id.: 6FYCY7524J5NNQE1011CEFA6VK40D9WP2G1X4G8YPOF4L24

cdomingo, 2018-06-20 08:56:09
 NiMMbus Id.: 06133154N3131LHXS18FIIMT82F8604IP417JUXUZ8D2J71
 Abstract: Pay attention!!
 Date (creation): 2018-06-20
 Rating: 5/5

NiMMbus

The integration of the NiMMbus plugin in a portal or website allows users to easily share their experience by adding comments to specific data sources. These comments are visible to everyone and promote communication and collaboration, thus creating in a constant feedback loop

MSP Challenge

Collaboration in future planning.

Marinomica

Marinomica allows seamless sharing and embedding of their (native) content into other external applications.

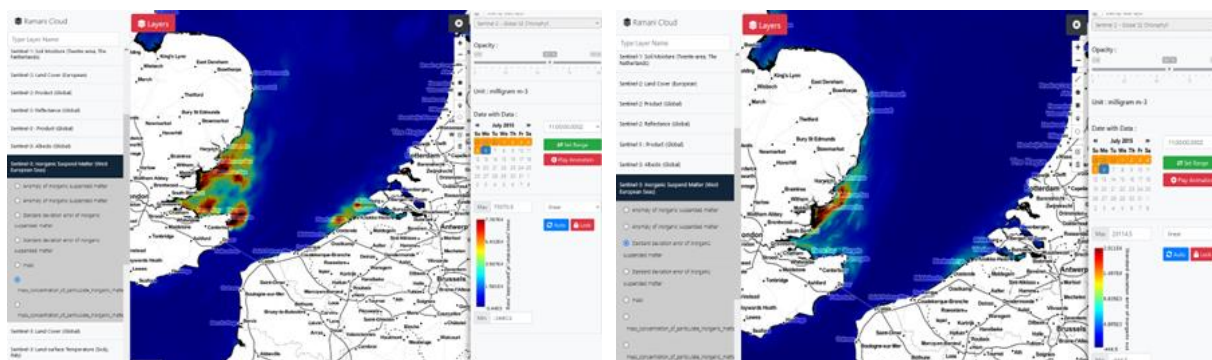
3.11 Distinguish and communicate the differences between historical and predicted data

When developing applications around geo-spatial data, one must consider whether the goal of the application is to monitor or to plan.

For monitoring, we can rely on historical and current data to simply assess and compare the changes. However, planning for the future requires predictive modelling, a statistical technique that uses mathematical and computational methods to predict a possible outcome. Predictive modelling analyses historical and current data in combination with other (external) factors, to create future estimations and approximations.

An interesting UI challenge that occurs when working with predictive models is: how to clearly communicate to the users what data is real and what data is an estimation. Furthermore, when uncertainty measures are available, it may be interesting to have a combined visualization schema for the estimated values and its uncertainty.

Examples:



Ramani

The Ramani application has 'standard deviation error' data layers available in their catalogue, that communicate the error rates on specific predicted data.

It could be beneficial to implement a function that allows users to select multiple data layers and visualize them together at the same time on the interactive map. This would allow for an easier side-to-side comparison of the predicted data and the error rates of said data.

3.12 Screen Size. Design for varying screen sizes.

Content should be developed to be viewed and interacted with across a range of screen sizes.

Firstly, not all functions and features make sense on all devices. It is important to identify the different scenarios in which an application will be used across the groups of devices that are supported. A suitable experience should be designed for each of those scenarios. For example, mobile users often require different things from a product than desktop users.

Secondly, different screens allow for different input methods. For example, interacting with a touch interface is not very precise, due to the (often) smaller screen size of touch devices and the use of our fingers to interact with the application, which is many times bigger than a cursor on a computer- or laptop screen. Elements generally need to be bigger and be spaced further away from each other, to allow users to easily interact with content despite varying finger sizes and avoid clicking multiple things at the same time.

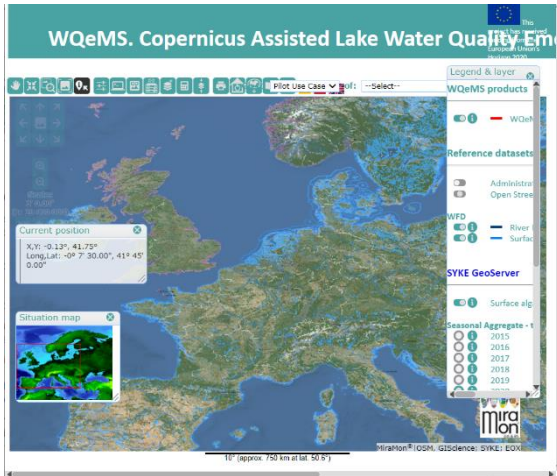
Even when an application is not meant to support a wide range of devices with extremely varying screen sizes, it is still an important aspect to consider. Modern computer- and laptop screens have a few different common sizes, but it can't be assumed that every user in the target audience possess a device with these popular specs.

Adapting UI to various screen sizes is often more complex than simply resizing content in the same ratios. Content hierarchy should remain clear; content shouldn't fall outside of the designated areas; there should still be sufficient breathing space etc. If done incorrectly, the clarity of the application can be very negatively influenced.

It is noteworthy that desktop applications that require to be downloaded and installed, can more easily demand specific specs to be run. Of course, this is not a solution as it doesn't make an application more accessible in itself. Ideally, the UI should still be designed in a way that supports varying screen sizes. However, it does allow to clearly communicate to the user whether their setup is optimal or not. This is harder to achieve on websites, which users can easily access within a web browser, making it is impossible to reinforce the use of a specific device. The use of desktop applications also has downsides, such as extending the required setup steps and time before an application is ready to be used. This is why the options should be considered case by case and will highly depend on the target audience.

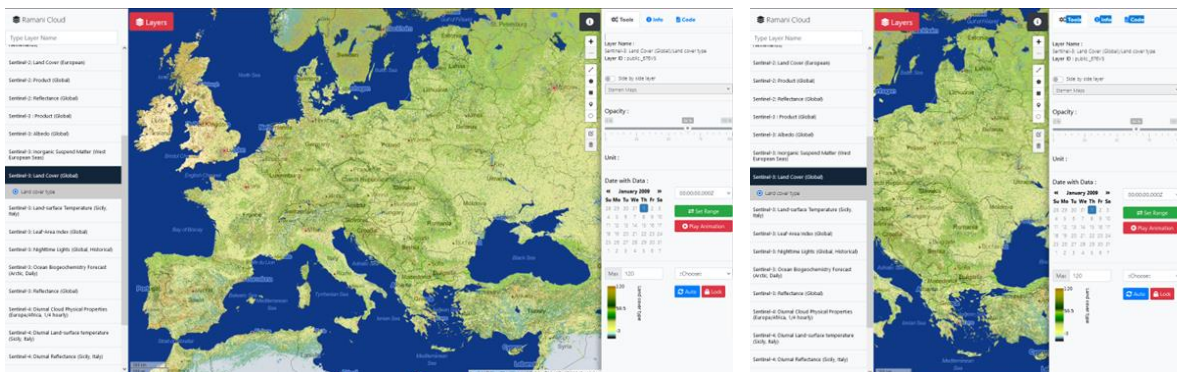
Examples:





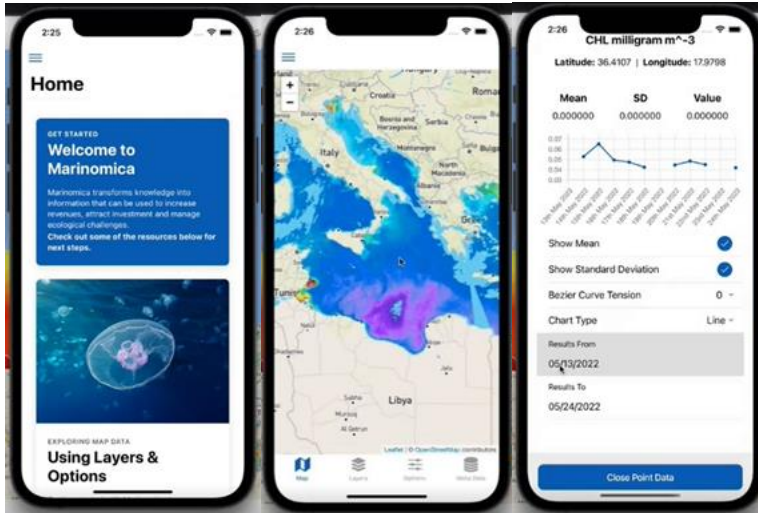
MiraMon Map Browser

While the MiraMon Map Browser has been designed to be used in full-screen mode on modern computer- and laptop screens, it is capable to adapt to alternative screen- and browser window sizes. Element placing and scaling is not always perfect, but the interface does remain clear and useable.



Ramani

Unlike the MiraMon Map Browser, the Ramani application doesn't scale the user interface accordingly to the screen size. The division ratio between the different sections (the catalogue on the left, the map in the middle and the data layer information on the right) adapts but none of these elements are actually scaled. This creates for an overcrowded and unbalanced interface on very small screens, as the catalogue and data layer sections remain the same size while the map section is greatly reduced.



Marinomica

It is noteworthy that Marinomica is available as a full-size browser version, as well as a mobile and tablet application. This makes it more accessible in general, as it supports a wider range of devices, but also allows users to interact with the application on the move, which might open up some additional use cases. However, not all the functionalities of the browser version are available in the later.

3.13 Avoid Data Duplication

While this lesson is not directly related to the design of front-end user interfaces, it definitely influences the user experience. It is something to consider for general efficiency.

In ILIAD, various complementary applications will be created that are bound to overlap in functionality and implement similar- or the same data sources. However, each application also has its own strong points and unique features that sets it apart. In the end, it will be up to the users to figure out for themselves what application fits their needs and use case the most, in the broadest sense of the word.

Though each partner application is its own, standalone product, the consistency of the data across the various applications will influence the usability and credibility of all of them.

It is important for the data to be consistent to avoid users trying out a few different applications that offer a digital twin of the ocean, only to find out that certain data values and time ranges vary. We want to avoid making a user wonder whether data is legitimate and which of the applications is 'right'.

4 Identified UX/UI Design Opportunities

4.1 VISTools Analytics

As the existing prototype application leverages the BI framework from VISTools, there is a good opportunity within the scope of Iliad to help craft a more refined UI, specifically tailored to the data and individual users.

The first stage of personalization would be **role-based** personalisation. Role-based personalisation in this case, for example, may mean a different UI for a business owner and the skipper of a vessel respectively. Different types of users could benefit from a simplified UI, showing only metrics of interest with customised user dashboards that refine the UX for these user types. Later iterations could also move into **individualised** personalization. Individualized personalization is fundamentally different to role-based personalization and is characterised by an individual UI/UX for each user, showing a different interface to each person based on preference.

The sensitive nature of the data in this application, could also benefit from an approach to UI that helps to instil trust. Data owners should feel comfortable that they will be the only ones who will benefit from the commercial advantages of certain data, signposting exactly what will remain privileged or what will be shared and with whom, subject to any prior agreements.

4.2 Marinomica

As this platform is primarily focussed on visualization, rather than hosting or cataloguing large amounts of data, the platform could be used to provide an instant UI for some pilot projects in leu of a specific MVP, or with modification form the basis of an entry point to pilot data. For example, Marinomica (or its successor GeoMachine) could connect to the API for the jellyfish swarm pilot, displaying that data alongside essential ocean variables such as sea temperature, helping to view correlations, in turn contributing to the development of a forecasting model.

Numerous pre-built UI patterns such as customisable user dashboards, data display components for time-series, map and contextual data amongst others, offer a very fast way to connect raw data to visualisations.

The 'machines' area of the platform is a way to create more focused environments using the same set of tools available in the main UI, allowing users to look at certain digital processes in isolation. This could be useful for creating dedicated areas for twins.

4.3 MSP Challenge Simulation Platform

Whilst the MSP challenge has a specific remit, its ability to explore the impact and management of the marine economy, ecology, and environmental policy in a deep and accessible way, makes it a great compliment to many of the pilot schemes.

Although ostensibly a game by design, it should be noted that gamification techniques are increasingly used as part of the UX for information retrieval systems. With its game mechanics

and multi-player offering, the MSP Challenge is uniquely positioned to make what can be a somewhat dry and inaccessible area into something more intuitive and engaging. The gamification elements demonstrated here should be considered in future UI / UX work as a powerful complimentary tool to the more traditional building blocks for content / data analysis.

Several studies have highlighted the different role of gamification between generations, and this should be a consideration moving forwards. “Generation X show that perceived usefulness and recognition are an essential aspect of gamification. While for millennials, they show that playfulness and social influence are important factors.” (Dyah Wahyu Sukmaningsih, 2020)

4.4 ADAM

Bringing together distributed data from different infrastructures and offering a unique 3D interface, this is a very different take on the fusion of data discovery and visualization, likely owning from its roots as a platform developed for the European Space Agency.

Much can be learnt from the way that various UI elements can be switched off within the ADAM interface, progressively simplifying the UI. Work in 6.2 should look to implement similar progressive simplification where possible, much like with different data dashboards in ILVO’s pilot, presenting UI alternatives, targeted at different audiences.

ADAM integrates with Jupyter Notebook. Data science is far from a mere ‘trend’ these days, cementing itself as a cornerstone of modern content analysis. This has been helped in part by tools such as Jupyter Notebook. Jupyter has quickly become the de facto standard for data specialists, allowing them to work more productively by greatly reducing the effort needed to process data and run variations.

4.5 Ocean Data Platform

The Ocean Data Platform is a bold and ambitious product, that aims to aggregate a huge range of ocean data sources as well as providing a level of data processing. As such, the platform occupies a unique position as one few project platforms with the ability to host pilot specific data as well as providing one-stop-shop for the analysis of cloud data from multiple sources.

Already featuring a marketplace of products directly derived from the platform data, there are obvious parallels in their approach to the marketplace that Iliad endeavours to build.

The Ocean Data Platform also integrates with Jupyter Notebook. The ubiquity of Jupyter Notebook within the marine research and data science community at large, makes it an obvious technology to attempt to integrate from a UX perspective. The integration of data tools such as Jupyter Notebook will be something that we should assess and continue to develop for other platforms where possible, talking directly to users about specific points of integration and ways to make data analysis easier.

4.6 Ramani

One of the most fascinating aspects of this platform is the ability to create new products and prototypes from inside the platform, using the richly annotated catalogue to create a new product that can be exported as code.

On-the-fly app creation is a complicated proposition in terms of UI / UX and we should look for opportunities to learn from the experiences of users with this platform. The ability to quickly iterate new applications may be of interest to pilots that don't yet have any pre-built UI.

The code generation / app infrastructure provided by Ramani could be a useful tool in bringing new concepts to market more quickly.

4.7 NextGEOSS

One of the focuses for the pilot areas is the creation of specific products and services for a future marketplace. Many ocean / geo-based products are inherently complex due to the broad array of technologies used and the fact that many of these products begin life as the result of research projects. To this end, NextGEOSS has dedicated support features to aid users.

The integration of user help and easily accessible support as demonstrated by NextGEOSS should form a key tenant of best practices for UI/UX development going forwards. Although ideally the perfect application should be intuitive enough to use without documentation, the rapid development and infancy of Iliad products (as well as the inherent complexity) will necessitate the need for well-structured support in the short to medium term.

4.8 NiMMbus

NiMMbus offers personalized options for the user, such as saving new combinations of data layers to the interface and is a good example of the sort of the personalisation that we should look to achieve in future UI / UX work.

Presently, it is the only system that specifically employs the Geospatial User Feedback (GUF) specification from the OGC. Feedback is a critical aspect of any product's evolution, and we should look to implement this standard or similar in future products. Consumers of geospatial data products are typically quite niche. Understanding the needs of these users through feedback will help to mature marketplace products and ultimately increase their commercial viability, so long as quality feedback can be generated that in turn leads to meaningful actions and iterations.

5 Conclusion

In this report we presented 13 lessons on UX/UI design particular to the analysis of eight base systems that partners have brought into the ILIAD project:

- 1. Consistency.** *Keep repeated content uniform.*
- 2. Minimalist Design.** *Prioritize and limit the amount of information and functions per screen or section.*
- 3. Streamline the User Journey.** *Organise (inter)actions into groups.*
- 4. Status Visibility & Feedback.** *Keep the user updated on what the application is doing and provide regular informative and relevant feedback.*
- 5. Help & Documentation.**
- 6. Ease of Use.** *Cater the experience to both novice- and experience users.*
- 7. Design for Accessibility.**
- 8. Allow Customizability.**
- 9. Search & Explore Function.**
- 10. Sharing & Collaboration.**
- 11. Distinguish and communicate the differences between historical and predicated data.**
- 12. Screen Size.** *Design for varying screen sizes.*
- 13. Avoid Data Duplication.**

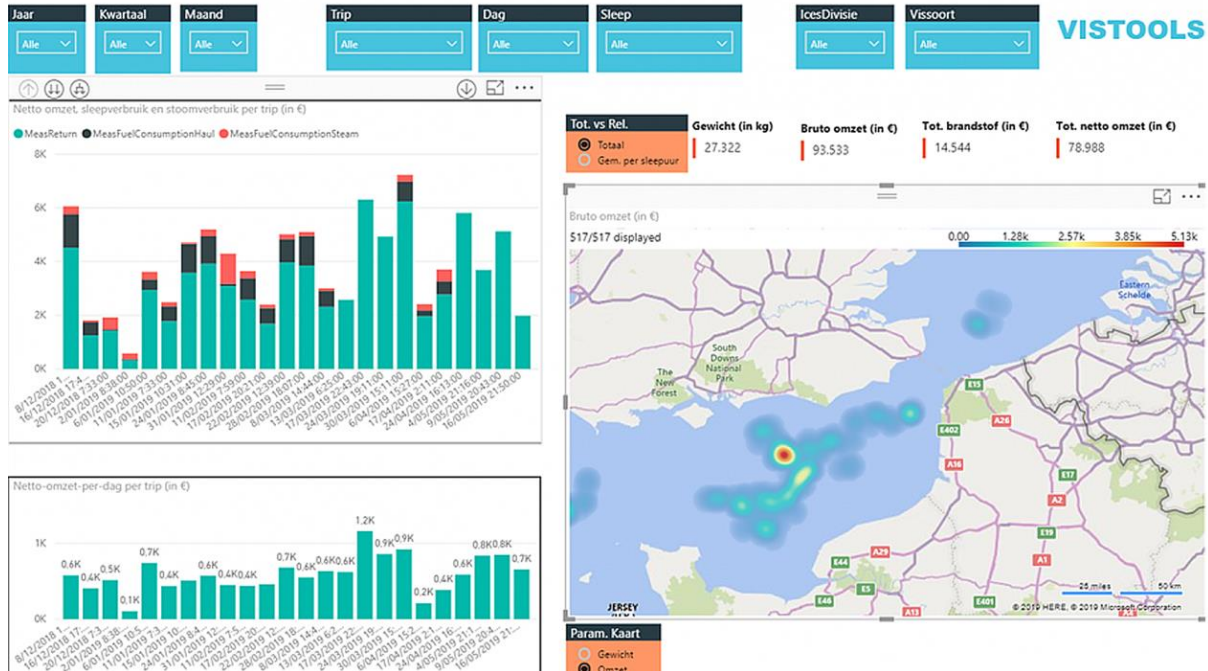
These lessons will be brought along as Work Package 6 as well as the ILIAD project more generally continue with their end-user experience and interface design and development efforts. More specifically, Work Package 6 will involve the design and development of end-user application prototypes focusing on:

- The Marketplace, and general user experience and interface elements (T6.2)
- Performance indicators and dashboards (T6.3)
- Geo-data visualisation (2D, 3D, 4D; T6.4)
- Immersive, Interactive, VR visualisation (T6.5)

In all these tasks, the involved partners will need to set up their design and development process in such ways that good UX/UI design can come into full fruition, particularly applying the aforementioned lessons.

6 Appendix – Partner Applications

VISTools Analytics – ILVO



Demo date: 25/05/2022

What is VISTool Analytics?

'VISTool Analytics is a Power BI interface used for real-time monitoring of fishing vessel and creates a feedback loop to provide the information back to the vessel owners.'

Who is the target audience?

Vessel owners.

What is the main use case / goal?

Currently, a lot of information such as fuel usage, the value of weekly catches, rich fishing grounds etc. is only available to the skippers and vessel owners after the trip has already been concluded.

By automating this process and giving as real-time possible feedback, the fishers can work-and use their resources more efficiently.

We note that this is sensitive data for these vessel owners, as it has clear commercial and competitive value.

Step by step walkthrough:

The experience will differ slightly depending on which dashboard the user chooses to consult. In the following example we will focus on the 'Revenue dashboard tab'.

1. At the time of writing, VISTools was not publicly accessible. Contact ILVO through <https://ilvo.vlaanderen.be> for more information.
2. Once you've opened the app, select a dashboard in the bottom tabs. (Revenue, Species & Location, Time, Fuel, Route etc.)
3. Select different 'slicers' in the top menu to filter the data by. (Time period, trip, fish type etc.)
4. Consult the graphs. (*Graph 1*: The estimate net revenue of every trip is represented by a bar, red represents the cost deducted while steaming, black represents the cost deducted while fishing. *Graph 2*: The estimate net revenue per day on average.)
5. Consult the interactive heat map on the right to view most successful fishing grounds.
6. Use the 'drill down' function to view deeper / more detailed levels of data. (Revenue per day within a trip etc.)

What knowledge is required to use the tool?

- Familiarity with terms used in the fishing industry, such as 'steaming'.
- Mastering the Dutch language. The current version doesn't support other languages.

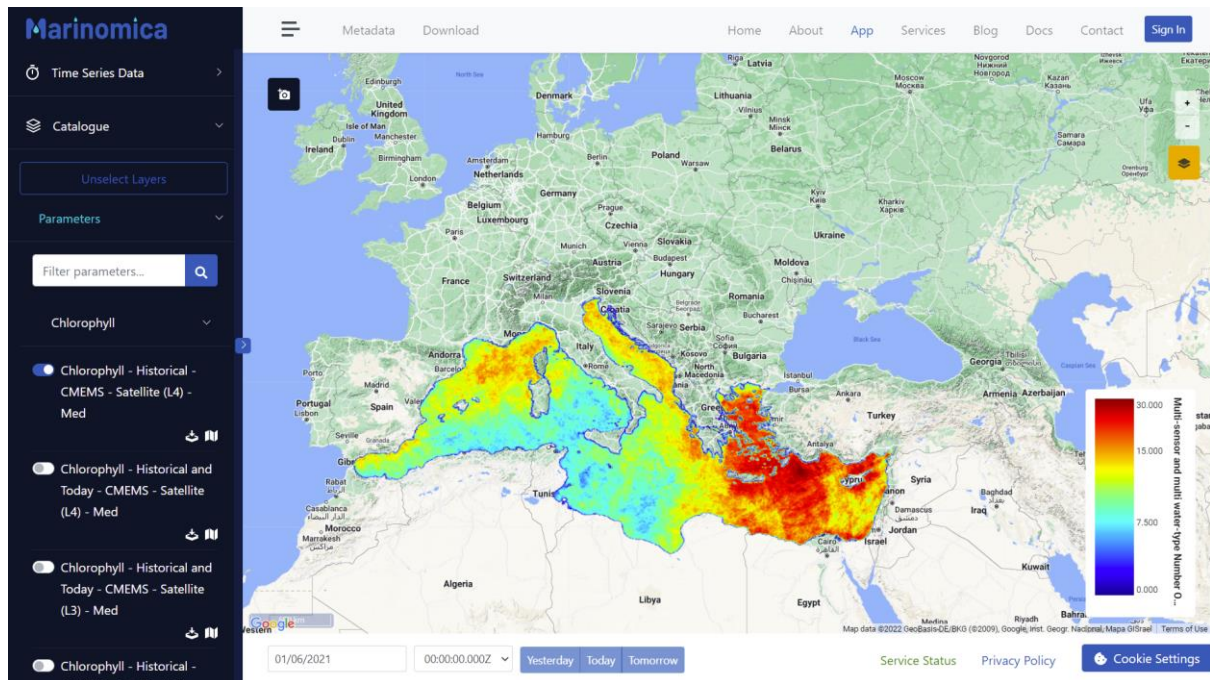
Functions / aspects similar to other partner tools:

- Web based application.
- Use of interactive (heat) map.
- Dashboard overview.
- Offers a variety of different dashboards focussed on different thematic data sets.

Unique functions / aspects not found in other partner tools:

- Sources the majority of the data from their end users.

Marinomica – Blue Lobster



Demo date: 25/05/2022

What is Marinomica?

'Marinomica is a web GI platform for the visualisation of on-demand data services, with predictive models and forecasts.'

Marinomica is no longer being actively developed and will be succeeded by GeoMachine. GeoMachine is a complete rewrite, utilizing a modern technology stack and a simplified architecture, with all of the features present in Marinomica as well as the added ability of running on-demand services.

These on-demand services or 'machines' are specific environments for running data models, allowing users to change the underlying variables and affect the outcome. These machines leverage the UI elements of the main catalogue, whilst focussing the toolset to a specific question or topic area.

All existing Marinomica data will soon be available in GeoMachine.

Who is the target audience?

Undefined.

What is the main use case / goal?

Undefined.

Step by step walkthrough:

1. Visit www.marinomica.com/map and log in.
2. Optional: Select a base map style.
3. Select one or more data sets of interest in the catalogue on the left, to visualise on the map. Use the search function to find certain data more easily / quicker.
4. Select a specific date or date range in the bottom bar. (Different data layers have different time dimensions (daily, weekly, monthly) and date ranges. Certain data layers are up to date and have forecast capabilities. Other data layers are historical making the options more limited.)
5. In the 'Map Layers Manager' window, toggle other layers within the data set on or off. The different layers can also be rearranged.
6. Optional: Customise the visualisation of the data layers on the map.
7. Consult the interactive heat map.
8. Select a point on the map to consult the properties for the data layers in that specific location. (Graph: Values across a period of time, edit curve tension, change graph type, edit the date range, export the information. Enlarge & zoom function to look at the data in more detail.
9. Optional: Consult the meta data of a data layer in the top bar.
10. Optional: Download aspects of the data set in the top bar.

What knowledge is required to use the tool?

- Familiarity with terms used in the data analysis industry.

Functions / aspects similar to other partner tools:

- Web based application.
- Use of interactive (heat) map.
- Dashboard overview.
- Dashboards can be exported.
- Data catalogue with both an explore and search function.
- Data layer visuals are editable. (Colour palette, opacity, depth etc.)
- Map base layer is editable. (Abstract / vector, satellite etc.)
- Availability of map grid overlay.
- No high-resolution ocean maps available.
- Editable time view.

- Panning controls by pressing the left mouse button and dragging simultaneously.
- Zooming in and out controls by scrolling the mouse wheel, using the UI buttons and double clicking on the map.
- Majority of UI elements can be hidden to have a more full-screen map view.
- Raw or processed data downloadable.
- All base maps can be equally zoomed in, even if certain don't have a more detailed zoom step available.

Unique functions / aspects not found in other partner tools:

- Windows specifically mention when they are draggable.
- Dashboards can be embedded.
- Export a screen capture.

We note that Marinomica will be succeeded by GeoMachine as the ILIAD project progresses. This platform extends beyond the original design of Marinomica to cater more effectively for interacting with digital twin data and simulations as well as doing everything that Marinomica did in a more robust interface.

MSP Challenge – BUAs



Demo date: 25/05/2022

What is MSP Challenge?

'MSP Challenge is a playful interactive simulation for the visualisation of data within a specific sea basin, development of future spatial plans and evaluation of the consequences of these decisions.' *(on energy, shipping and the marine ecology.)*

Who is the target audience?

Decision-makers, stakeholders and students.

What is the main use case / goal?

To help decision-makers, stakeholders and students understand and manage the maritime economy and marine ecology / environment.

Step by step walkthrough:

1. Visit www.community.mspchallenge.info/wiki/Download and log in.
2. Download and instal MSP Challenge.
3. Connect to a server and log onto a sessions of a specific ocean area. (North Sea, Baltic Sea, Adriatic Sea and Clyde Marine Region.)

4. Select one or more data sets of interest in the catalogue on the left, to visualise on the map.
5. Consult the interactive map to get an understanding of the current state of the sea basin.
6. Discuss your ideas with your team members and set objectives. Consider what changes would be required to reach these objectives.
7. Consult the 'impact tool' to view the relationships between human activities and the marine ecosystem, and estimate the possible consequences of your ideas.
8. Create one or more spatial plans with the necessary data layers and implementation date.
9. Apply changes to the different data layers within a spatial plan by (re)moving, editing and adding new geometry.
10. Get the spatial plan approved. (If applicable: by other implicated / affected teams.)
11. Once a spatial plan has been implemented, assess the consequences. This can be done by comparing the visual changes in the data layers on the map before and after the implementation data, or by reviewing the changes in the key performance indicators (KPIs) in the KPI tabs or the objectives monitor.

What knowledge is required to use the tool?

- (Basic) knowledge on maritime spatial planning.
- Familiarity with terms used in the maritime spatial planning industry.

Functions / aspects similar to other partner tools:

- Use of interactive (heat) map.
- Dashboard overview.
- Data catalogue with an explore function (search function coming soon).
- Availability of map grid overlay.
- No high resolution ocean maps available.
- Editable time view.
- Panning controls by pressing the left mouse button and dragging simultaneously.
- Zooming in and out controls by scrolling the mouse wheel or using the UI buttons.
- Majority of UI elements can be hidden to have a more full-screen map view.
- Coming soon: Help buttons. Shows info about the function of different elements.
- Availability of elaborate knowledge base.

- Availability of step by step (video) tutorial walkthrough of various functions.

Unique functions / aspects not found in other partner tools:

- Application requires to be downloaded / not a web-based application.
- Availability of UI scaling.
- Availability of a tool (impact tool) that shows the relationship between human activities and the marine ecosystem.
- Possibility to apply changes to data layers and evaluate the consequences.
- Possibility to turn text overlays on the map on and off.
- Possibility to turn different legend values within the same data set on and off separately.
- Multi-player engine which supports interaction and communication between users.

ADAM – MEE0



Demo date: 31/05/2022

What is ADAM?

'ADAM is a referencing implementation of what in the OCG ecosystem we call a data cube.'

'ADAM is providing a seamless access service to data distributed across different infrastructures.'

Who is the target audience?

Initially, the European space agency or the data providers in general.

This has evolved and expanded a bit over time.

What is the main use case / goal?

The scope was 'how to make the data findable, accessible, interoperable and reusable'.

Step by step walkthrough:

1. Visit <https://explorer.adamplatform.eu/> and log in.
2. Optional: Select a base map style.
3. Select one or more data sets of interest in the catalogue on the left, to visualise on the map. Use the search function to find certain data more easily / quicker.
4. Select a specific date or date range.
5. Optional: Customise the visualisation of the data layers on the map.

6. Consult the interactive heat map.
7. Select a point on the map to consult the properties for the data layers in that specific location.
8. Use the Jupyter Notebook to access the analytic / computing component.
9. Optional: Download aspects of the data set in the top bar. / the data of selected area.

What knowledge is required to use the tool?

- Knowledge of coding when using the back-end interface Jupyter Notebook.

Functions / aspects similar to other partner tools:

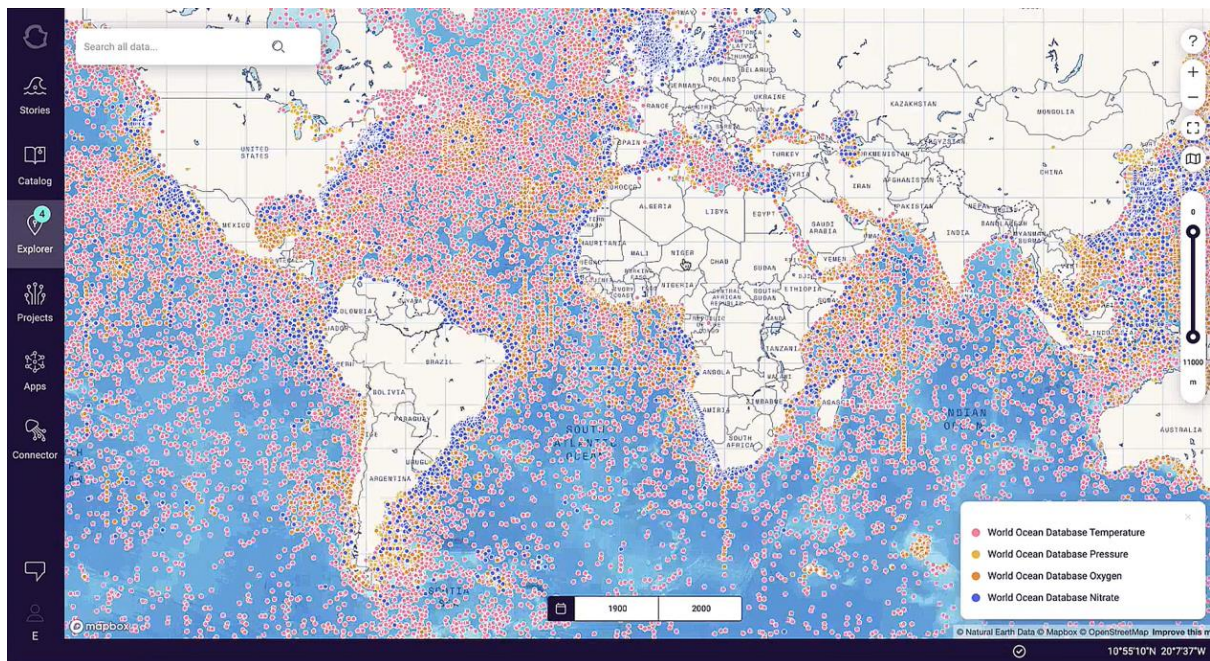
- Web based application.
- Use of interactive (heat) map.
- Dashboard overview.
- Dashboards can be exported.
- Data catalogue with both an explore and search function.
- Data layer visuals are editable. (Colour palette, opacity etc.)
- Map base layer is editable. (Abstract / vector, satellite etc.)
- Availability of map grid overlay.
- No high resolution ocean maps available.
- Editable time view.
- Panning controls by pressing the left mouse button and dragging simultaneously.
- Zooming in and out controls by scrolling the mouse wheel or using the UI buttons.
- Majority of UI elements can be hidden to have a more full-screen map view.
- Raw or processed data downloadable.
- Data doesn't get duplicated, it remains in its original location.
- Availability of Jupyter framework, live-coding back-end interface.
- All base maps can be equally zoomed in, even if certain don't have a more detailed zoom step available.

Unique functions / aspects not found in other partner tools:

- Globe view.
- 3D visualisation mode of the map.

- Availability of pixel based access. Capable to access / define data within a pixel, without loading the whole data base. A specific request gets sent.
- The cursor movement on the map shows live coordinates.
- Only allows to process data using the Jupyter framework, not through the front-end web application.

Ocean Data Platform – HUB Ocean



Demo date: 21/06/2022

What is Ocean Data Platform?

'An open collaborative tool built in Mapbox, that liberates and aggregates data from different sources (public and private) in one place where people can easily access, consult- and build off of it.'

Who is the target audience?

Mainly scientists in both academia and industry.

Citizens and civil society.

Governments and policy makers.

What is the main use case / goal?

The goal is to aggregate data from different sources into an accessible and comprehensive environment, to encourage scientific collaboration, industry transparency and regulatory power-all with the goal of creating a healthy and productive ocean.'

Step by step walkthrough:

1. Visit <https://oceandata.earth/> and log in.
2. Optional: Select a base map style.
3. Optional: Visit the 'Stories' tab.

4. Select one or more data sets of interest in the catalogue, to visualise on the map.
Use the search function to find certain data more easily / quicker.
5. Visit the 'explorer' tab to visualise the selected data layers on the interactive map.
6. Optional: Customise the visualisation of the data layers on the map.
7. Select a point on the map to consult the properties for the data layers in that specific location.
8. Optional: Consult the 'projects' tab. Save (thematic) data sets and specific settings into collections, to be able to easily access, edit and share the information at a later point.
9. Optional: Consult the 'apps' tab. Pick one of the existing options. (Lusedata, Ship Emissions Simulator, Ship Emissions tracker.)
10. Optional: Consult the 'connector' tab. Aggregated data from public and private sources, made accessible in the cloud and ready to be manipulated in Jupyter framework. Also supports API and SDK (Python and JS).

What knowledge is required to use the tool?

- Knowledge of coding when using the back-end interface Jupyter notebook.

Functions / aspects similar to other partner tools:

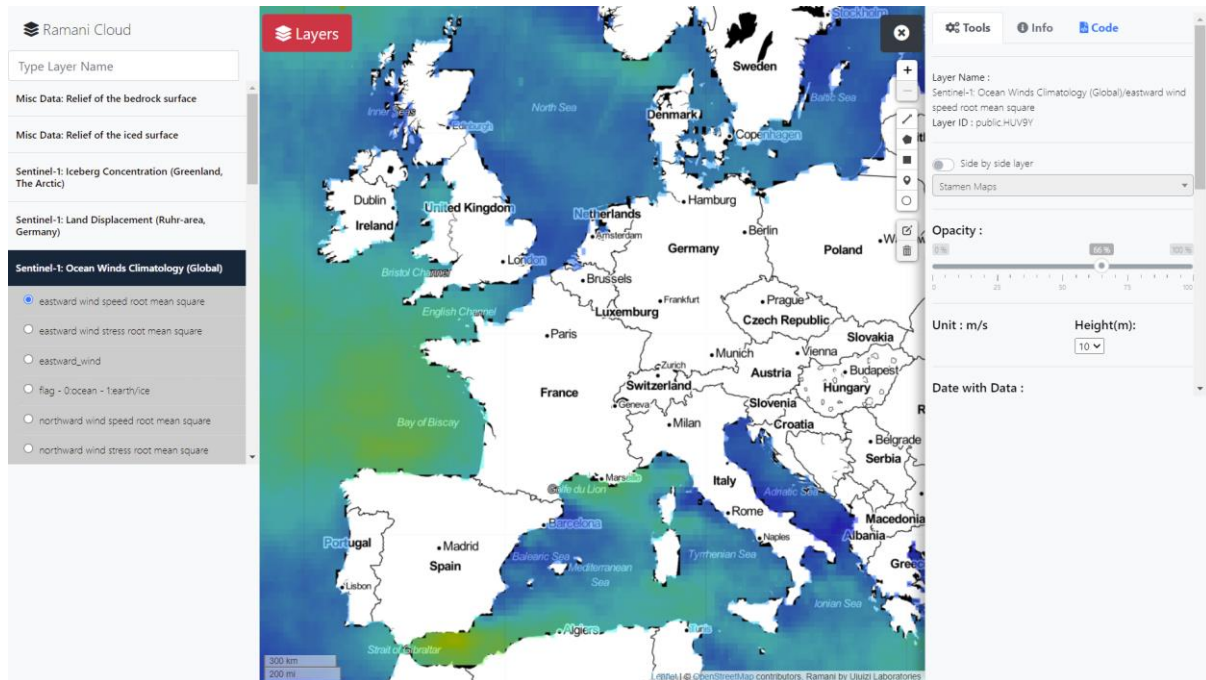
- Web based application.
- Use of interactive (heat) map.
- Dashboard overview.
- Dashboards can be exported.
- Data catalogue with both an explore and search function.
- Data layer visuals are editable. (Colour palette, opacity, depth etc.)
- Map base layer is editable. (Abstract / vector, satellite etc.)
- Availability of map grid overlay.
- No high resolution ocean maps available.
- Editable time view.
- Panning controls by pressing the left mouse button and dragging simultaneously.
- Zooming in and out controls by scrolling the mouse wheel or using the UI buttons.
- Majority of UI elements can be hidden to have a more full-screen map view.
- Raw or processed data downloadable.
- Data doesn't get duplicated, it remains in its original location.
- Availability of Jupyter framework, live-coding back-end interface.

-
- Help button. Shows info about the function of different elements.

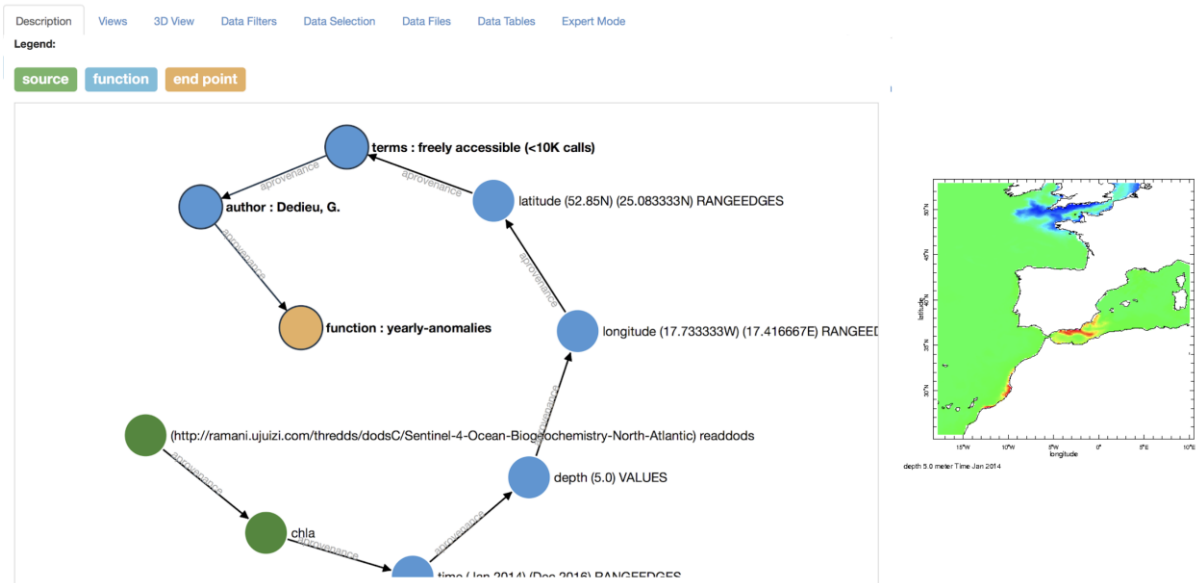
Unique functions / aspects not found in other partner tools:

- Availability of `stories`. A more article-like, accessible and visual approach to communicating the context behind data.
- Availability of `projects`. Save (thematic) data sets and specific settings into collections, to be able to easily access, edit and share the information at a later point.
- Availability of `apps`. Connecting a network of (partly third party) applications easily created with specific (thematic) ODP data, providing a dashboard-like overview for the data.
- Availability of non-gridded data.
- Aims to be both a data profiler and ingest data entries from the users.
- Coming soon: show devices that provide data on the map. When selecting, it shows the animated 3D model. Possibly real time data updates.

Ramani – Ujuizi Lab



Processing workflow implementation; example of lineage and acknowledgments



Demo date: 22/06/2022

What is Ramani Services?

RAMANI Analytics is an online data facility to accelerate added-value products co-creation from Copernicus data, amongst others. With [+300 datasets](#) intuitive UIX and some [+100 algorithms](#), processing workflows can be shared between thematic and digital experts to ease collaboration and co-creation efforts. To simplify uptake and capacity building, already [+40 existing online tutorials](#) teach the use of big data analytics in bringing decision-ready info to accelerate Citizen Sciences engagement.

RAMANI Cloud is an online data publishing and 2-way crowdsourcing facility, where workflow outputs from RAMANI Analytics may also be directly consumed in smartphone Apps. Using [RESTful or native Map-SDKs](#) (iOS, Android, Web), allowing decision-ready information to be consumed where choices are made (farmer in the field, fisherman at shallow seas, buyer at the fish market, etc.), till date +65 crowdsourcing Apps have been developed over +6 years at some of Europe's largest innovation competition events.

Who is the target audience?

Decision makers, App developers, Geo-scientist interested in adding-value to standard Copernicus data collections

What is the main use case / goal?

Bring analysis-ready information to decision-makers in the field (a farmer in the field, a fisherman in shallow seas, etc.).

Step by step walkthrough:

1. Visit www.ramani.ujuizi.com and log in.
2. Explore the spatial data on offer in the Streaming Data Library (SDL).
www.analytics.ramani.ujuizi.com
 - a) By [source](#)
 - b) By [category](#)
 - c) By searching
 - d) By [exploring](#) through the front-end web-based map viewer.
 - e) By example through maps created by peers.
3. Visit the front-end web-based map viewer. www.maps.ramani.ujuizi.com
4. Select a data set of interest in the catalogue, to visualise on the map. Use the search function to find certain data more easily / quicker.
5. Optional: Customise the visualisation of the data layers on the map, in the tool tab on the right.
6. Optional: Select a date or date range to view the data at.
7. Place a marker on the map or draw geometry to consult the properties of the data layer in that specific location. Also available in json (JSON-LD) format.
8. Consult information about the activated data set (in the info tab on the right).
9. Optional: Add your own maps / data sources or link to opendoc URL. (Upload grid files through the editor and style the layer(s) of your own or existing maps through the Class Wizard.
10. Optional: Publish your spatial content to mobile or web using Ramani Maps-API.

What knowledge is required to use the tool?

- Knowledge of coding when using the back-end interface Jupyter notebook.

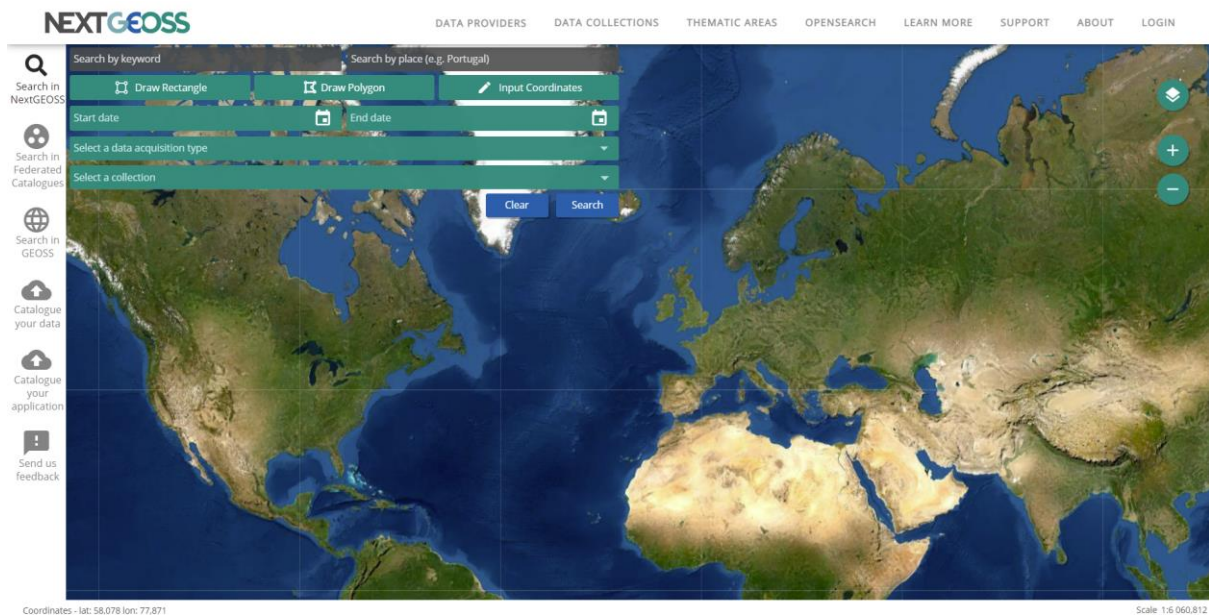
Functions / aspects similar to other partner tools:

- Web based application.
- Use of interactive (heat) map.
- Dashboard overview Data catalogue with both an explore and search function.
- Data layer visuals are editable. (Colour palette, opacityetc.)
- No high-resolution ocean maps available.
- Editable time view.
- Panning controls by pressing the left mouse button and dragging simultaneously.
- Zooming in and out controls by scrolling the mouse wheel or using the UI buttons.
- Majority of UI elements can be hidden to have a more full-screen map view.
- Raw or processed data downloadable.
- Data doesn't get duplicated; it remains in its original location.
- Availability of Jupyter framework, live-coding back-end interface.
- Availability of elaborate knowledge base.
- Availability of step-by-step tutorial walkthrough of various functions.
- Availability of [helpdesk](#).

Unique functions / aspects not found in other partner tools:

- Fast prototyping services using the data sets.
- Ability to consume data using IOS, Android, Javascript, Node JS, SAP-HANA, API reference.
- Third parties can make workflows.
- Availability of split screen / side by side layer comparison.
- When a data layer is selected that has data only applying to a specific (smaller) area, the map automatically zooms into / centers it.
- Map drawing tool to select an area, automatically generates a visual graph and json format code for the data from the selected area.
- Markers can be placed on the map.

NextGEOSS – Deimos



Demo date: 23/06/2022

What is NextGEOSS?

NextGEOSS is a centralised European Earth observation data hub and platform, that allows for easy discovery of- and accessibility to various data sources.

Who is the target audience?

Scientific communities and universities.

What is the main use case / goal?

The concept revolves around providing the data and ICT resources needed, together with the cloud services, seamlessly connected to provide an integrated ecosystem to support the deployment of Earth observation-based applications and services.

Step by step walkthrough:

1. Visit www.catalogue.nextgeoss.eu and log in. (To have full access to all functionalities.)
2. Optional: Select a base map style.
3. Explore the NextGEOSS catalogue. Use the search function with various filters to narrow down the data sets (search by keyword, place, coordinates, date range, data acquisition type and collection.) Or search for data sets available in a specific area by drawing geometry on the map.

4. Consult the meta data of one or more data sets of interest in the catalogue or by selecting the visualised area on the map.
5. Optional: Download the data set or other resources from SciHub.
6. Optional: Add user feedback.
7. Explore other European catalogues federated by NextGEOSS. (The following search function is pretty much identical to the NextGEOSS catalogue described before.)
8. Explore the GEOSS (data core) catalogue. Use the search function with various filters to narrow down the data sets.
9. Optional: Explore data sets per provider or collection by visiting the first two tabs in the top menu bar. Use the different categories on the left of the pop-up to narrow down the data sets.
10. Optional: Explore thematic areas by visiting the third tab in the top menu bar. A collection of NextGEOSS DataHub catalogues pilot services developed and integrated within NextGEOSS and the services developed by other European Projects across several thematic areas. (Example: agriculture, air pollution, biodiversity, energy, food security etc.)
11. Optional: Consult the machine to machine back-end interface, OpenSearch API
12. Optional: Catalogue your own data by sending a ticket creation request through the service desk.

What knowledge is required to use the tool?

None.

Functions / aspects similar to other partner tools:

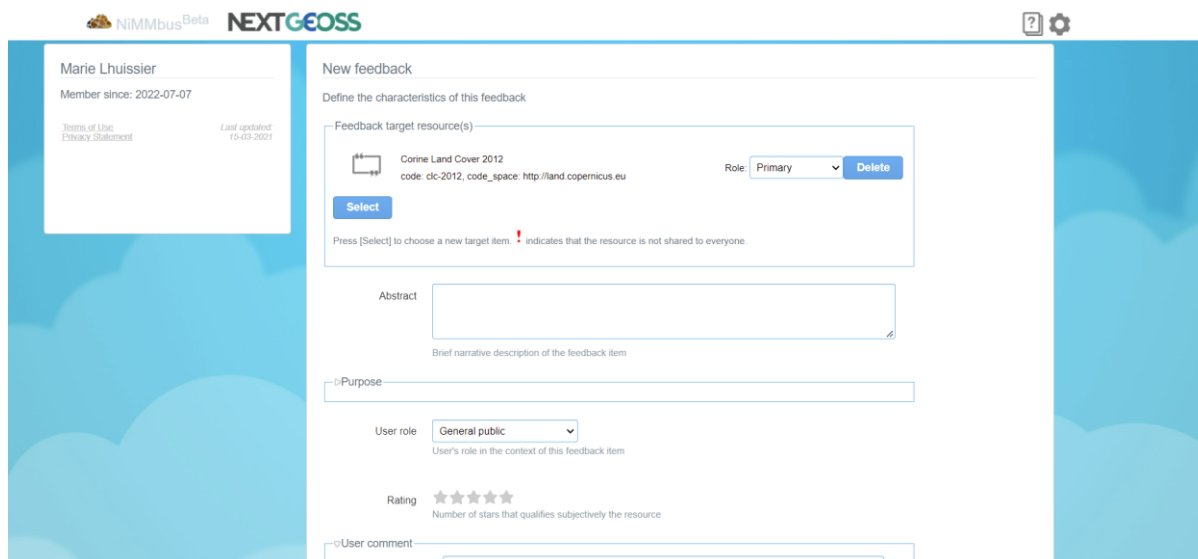
- Web based application.
- Use of interactive (heat) map.
- Data catalogue with both an explore and search function.
- Map base layer is editable. (Abstract / vector, satellite etc.)
- No high resolution ocean maps available.
- Editable time view.
- Panning controls by pressing the left mouse button and dragging simultaneously.
- Zooming in and out controls by scrolling the mouse wheel, using the UI buttons and double clicking on the map.
- Raw or processed data downloadable.

- Data doesn't get duplicated, it remains in its original location.
- Availability of service / help desk.
- No high resolution ocean maps available.
- All base maps can be equally zoomed in, even if certain don't have a more detailed zoom step available.

Unique functions / aspects not found in other partner tools:

- When a data set is offline, attempting to download it will automatically send an access request and the content should be back online within 24 hours.
- Focusses on distributing not visualising or processing data.
- Highest amount of available base map styles.
- Availability to discover all the data sets available in a specific area by drawing geometry on the map.
- When zooming in too closely into a map that doesn't have detailed enough visuals, there is a warning 'map data not available yet'
- All base maps can be equally zoomed in, even if certain don't have a more detailed zoom step available.

NiMMbus – Universitat Autònoma de Barcelona



Demo date: 07/07/2022

What is NiMMbus?

The NiMMbus service ingests user feedback about existing geospatial resources following [OGC Geospatial User Feedback \(GUF\)](#) Standard. It allows users to provide comments, ratings, questions, quality reports and publications related to a specific data set. NiMMbus can be used on its own but is more likely to be integrated in existing interfaces or catalogues.

Working implementations:

www.eneon.net/graph-ev-sdg

www.maps.ecopotential-project.eu

<https://catalogue.nextgeoss.eu/>

Who is the target audience?

Active GIS and RS data users that work with the data and have time to come back to the system and share experiences with data.

What is the main use case / goal?

Creating and sharing information and knowledge created by the users through Geospatial User Feedback system.

Step by step walkthrough:

From the user point of view:

1. Visit a geospatial resource
2. Find the button for adding feedback
3. See the www.nimmbus.cat/#front-page open and log in.
4. Wait to see the citation of the resource being created.
5. Provide feedback and accept.
6. See the feedback added to the list of existing feedback for this layer.

From the developer point of view:

7. Integrate NiMMBus in existing interface or catalogue with a JavaScript API and widget (Example: www.nimmbus.cat/test_widget.htm) or Javascript for the Web API (Example: www.nimmbus.cat/test_new_feedback.htm). The provided references describe the necessary steps, and the description of all the integration options can be found in https://github.com/grumets/nimmbus/tree/master/GUF_integration.

What knowledge is required to use the tool?

Knowing what a data layer is. Knowing how to download and operate with data in order to extract some information that can be of value for other people.

Functions / aspects similar to other partner tools:

- Web based application.
- Availability of knowledge base.
- Function to implement user feedback and observation data (similar to a feature coming out soon on Ocean Data Platform).
- Function to implement user feedback in the GEOSS portal: www.geoportal.org (the model is simpler).

Unique functions / aspects not found in other partner tools:

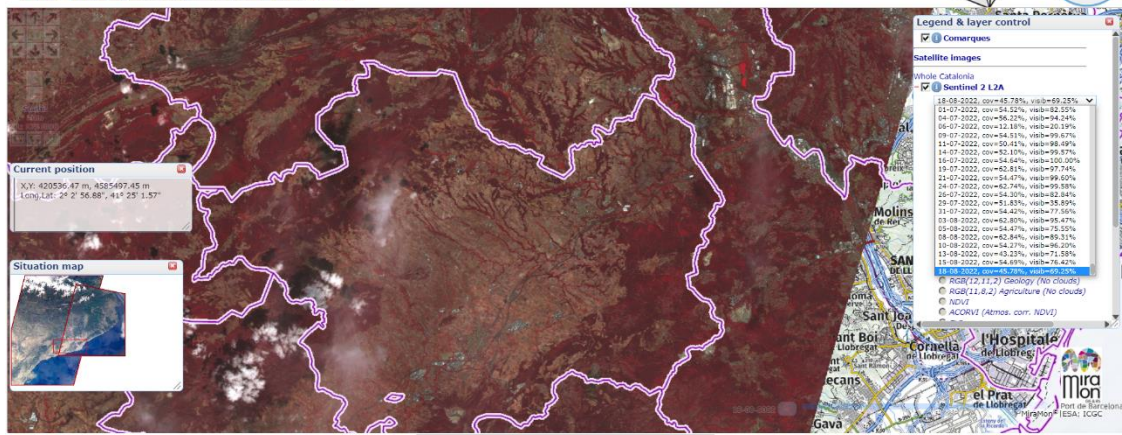
- The user feedback is in situ, with the relevant data set available as context, instead of sending the information to an external place where it needs to be reviewed separately.
- The main goal of the application is knowledge sharing and collaboration.
- The application can be integrated in other interfaces and catalogues.
- Feedback follows an OGC standard data model.

MiraMon Map Browser – CREAM

SatCat 2.0: Catalan Data Cube



UTM31N - WGS84



Demo date: 07/07/2022

What is MiraMon Map Browser?

The MiraMon Map Browser is a generic map viewer that provide the client side of OGC standards services to visualize geospatial resources as well as data such as GeoTIFF (COG) and GeoJSON. It allows users to view, download combine, animate and analyse geospatial data. The MiraMon Map Browser can show data from existing services and storymaps can be written about the data.

Working implementations:

www.maps.ecopotential-project.eu

<https://www.ogc3.grumets.cat/wqems/>

<https://www.datacube.cat/cdc>

Who is the target audience?

Any person interested in viewing geospatial services.

What is the main use case / goal?

Special focus is done on remote sensing time series data such Sentinel 2 data.

Test simple analytics before doing it directly on the data at full resolution.

Share

Step by step walkthrough:

1. Visit the page
2. Select the desired area
3. Activate the right layers
4. Create a new remote sensing index as a combination la layers and incorporate it as a new style
5. Share the created style to the community
6. Close the page
7. Reopen the page
8. Select the right layer
9. Retrieve a shared style
10. See it available in the list of styles

What knowledge is required to use the tool?

Notions on GIS basic analytics and RS basic band combination and indices

Functions / aspects similar to other partner tools:

- Web based application.
- Show remote sensing imagery
- Overlay of different layers in a single view

Unique functions / aspects not found in other partner tools:

- Animation of RS scenes
- Define RS indices and RGB combinations that are calculated on-the-fly
- Layer algebra calculated on-the-fly
- Sharing RS indices to everybody else
- Statistics and graphics of the data on the screen
- Connection to NiMMbus

References

Reference	Name of document
[REF-01]	ILIAD website. https://www.ocean-twin.eu/ . Retrieved 2022-03-22.
[REF-02]	Morville, Peter (2004, June 21). <i>Semantic Studios</i> . Retrieved July 13, 2022, from www.semanticstudios.com/user_experience_design