FP7 Project ASAP

Adaptable Scalable Analytics Platform

D6.5 Usability Report

WP 6 – Information Visualization
webLyzard technology

Nature: Report
Dissemination: Public

Version History

Version	Date	Author	Comments
0.1	27 Jan 2017	A. Scharl	Initial Version
0.2	17 Feb 2017	W. Rafelsberger	Component Usability
0.3	27 Feb 2017	A. Scharl	Review and Revision
0.4	07 Mar 2017	W. Rafelsberger	SUS Evaluation Results
1.0	08 Mar 2017	A. Scharl	Revision and Final Edits

Acknowledgement

The ASAP project has received funding from the European Union's 7th Framework Programme for research, technological development and demonstration under grant agreement number 619706.

Executive Summary

Deliverable D6.5 summarizes the work conducted in Task T6.4 of the ASAP project. It includes documentation of the chosen software design process in terms of usability, and reports the results of the usability and rendering performance evaluations.

The deliverable describes the approach taken to make usability considerations part of the core workflow of WP6 when developing individual information visualization services as well as the synchronized ASAP Visual Dashboard – focusing on components that support the rendering of data streams from dynamic sources from the two use cases based on *Web content* (WP8) and *telecommunications data* (WP9).

Following an iterative systems development approach, rapid feedback cycles and agile software development have been instrumental in the conceptualization and implementation of the ASAP dashboard. The deliverable summarizes the usability considerations for individual components by documenting major design decisions.

To assess the perception of the final dashboard version, we used an online questionnaire including a series of open-ended questions and a 10-item *System Usability Scale* (SUS) assessment. An overall SUS score of 74.5 reflects a favorable perception of the interface. Individual comments by the participants mentioned the complexity of the ASAP dashboard, but pointed out that it is fast and easy to use after appropriate training, while offering a rich and comprehensive feature set.

Table of Contents

Executive Summary **2**
Introduction **4**
Usability Evaluation Methodology **5**
Usability Heuristics 6
Qualitative Validation 7
System Usability Scale (SUS) 9
Dashboard Performance 10
Application of Usability Heuristics **10**
Drill Down Sidebar 10
Trend Chart 11
Temporal Controls 12
Geographic Map 13
Adaptive Tooltips and Context Menus 15
Summary **17**
References **18**

Introduction

The deliverable D6.5 summarizes the work conducted in task T6.4 of the ASAP project and includes documentation of the chosen software design process in terms of usability considerations and includes the results of the usability evaluations.

The overall goal of WP6 was to build information visualization (InfoViz) services embedded into an analytics dashboard to interactively explore web content (WP8) and telecommunications data (WP9), embracing both unstructured (Web intelligence) and structured (linked data) sources. T6.4 evaluated the usability of the work done up to T6.3, which included integrating the developed *InfoViz Services* into the unified ASAP Dashboard, shown in Figure 1.

The deliverable provides an overview of the used usability methodologies and describes how these were applied to different components developed for the ASAP Dashboard, including a documentation of major design considerations that guided the interface iterations.

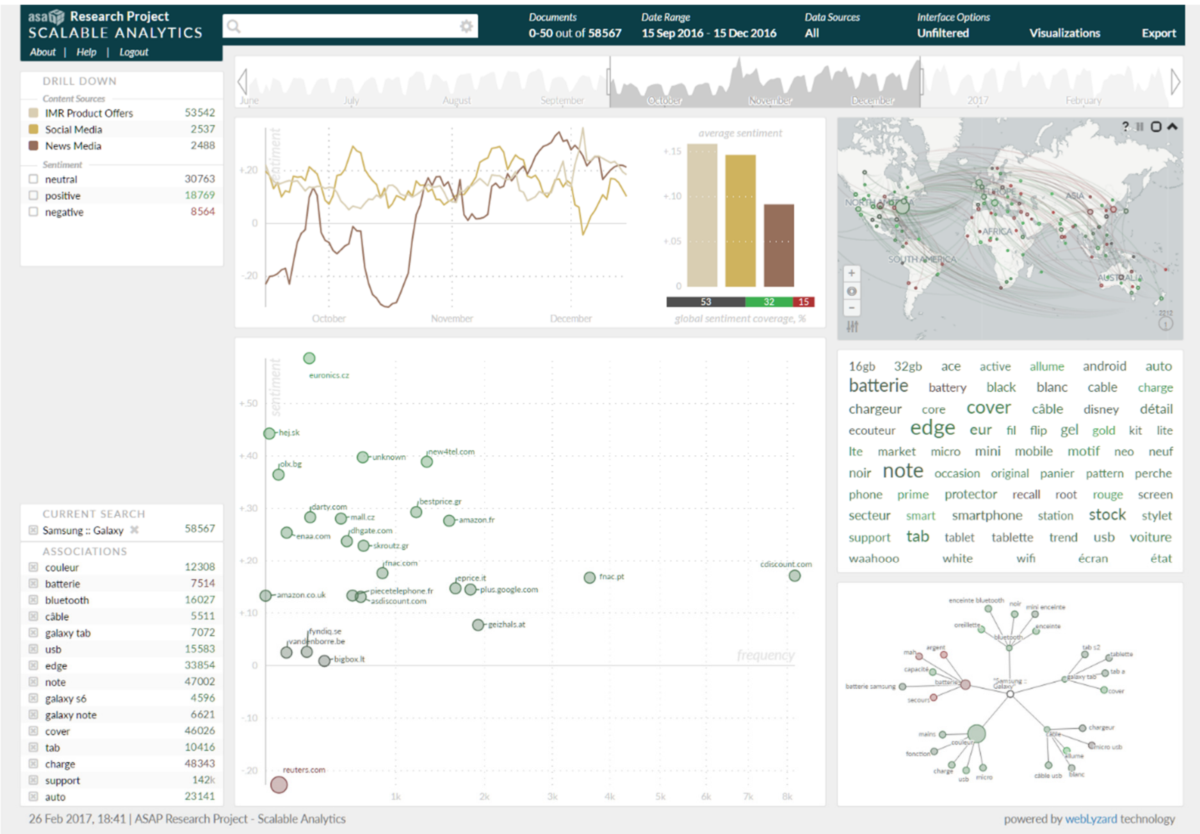

Figure 1. Screenshot of the ASAP Dashboard including the new timeline selector, line chart, bar chart, scatter plot and geographic map with aggregated location statistics

Usability Evaluation Methodology

To gain insight into the user experience of the ASAP Dashboard[1] following the initial deployment, usability evaluations were conducted. The aim of these evaluations was to determine strengths and weaknesses of the usability of the interaction design and was done by using different types of assessments:

- *Usability inspections* investigated the interface design against recognized usability principles ("heuristics"). These inspections were performed periodically during the design and implementation phases, so that improvements could be integrated into the prototype early in the development cycle;
- *Qualitative validation* of stable interface releases with small groups regarding subjective aspects such as expectations and satisfaction with the user experience, in line with the project's user-centered design approach;
- *System Usability Scale (SUS)*, a technology-independent questionnaire for measuring perceptions of usability (Brooke, 2013).[2]

The inspections and qualitative validations were done in iterative cycles depending on the project status: Following a user-centered design approach, usability considerations were part of the project throughout its development, and feedback loops were included in the manner of computational information design (Fry, 2004). During the implementation phase, *usability inspections* and *qualitative validation* were integrated into the development process. Finally, the *System Usability Scale* (SUS) questionnaire was used to gather user feedback.

Feedback from other initiatives such as the FP7 Project *DecarboNet*[3] or early adopters including the *National Oceanic & Atmospheric Administration* (NOAA) and the *United Nations Environment Programme* (UNEP) showed that test users have little difficulty using multiple coordinated views after receiving proper training. They appreciate the synchronized views to keep track of the semantic and geospatial context of their current tasks - e.g., the capabilities to structure the evolving public discourse, the visual identification of connections and trends for certain keywords, or the on-the-fly definition of categories and complex search queries via the topic editor. For untrained first-time users, however, the complexity of a dashboard can be overwhelming. Many different components and can be daunting for new users without a technical background.

---

[1] asap.weblyzard.com
[2] www.uxpajournal.org/sus-a-retrospective
[3] www.decarbonet.eu

Therefore, in addition to fulfilling requirements in terms of functionality, WP6 put special emphasis on ease of use and usability of the platform and its components, including considerations to keep complexity at a minimum when implementing new features.

Usability Heuristics

The ASAP Dashboard is a Web-based *Single Page Application* (SPA) following a multiple coordinated view approach (Hubmann et al., 2009). Its goal is to provide an experience similar to that of a desktop application. Nevertheless, certain usability considerations need to be taken into account given that the application runs in the context of a Web browser window.

Native desktop and mobile applications have varying methods for handling interactions depending on the operating system and device, e.g. which actions get triggered with a double-click, right mouse click, keyboard shortcuts or multi-touch gestures. Web sites, by contrast, often use hovering elements to support navigation. Browser applications themselves combine these paradigms.

For the average user, these different contexts are not always easy to comprehend. Creating Web-based SPA applications thus differs not only in technological terms from classic Web development, but also represents a challenge in regards to considering various overlapping interface paradigms depending on the user's context.

Since the *ASAP Dashboard* offers a rich feature set and supports multiple visualizations within a single application view (Scharl et al., 2016), we had to carefully make decisions to not make the user experience overly complex. In order to keep a consistent interface, we defined specific usability heuristics for the dashboard. Inconsistent usage of interface paradigms or usage patterns increase complexity and irritate users. While some of the following usability heuristics might seem straightforward at first glance, the actual challenge for a software designer is to consistently apply them throughout the design and implementation phase. For the user, providing a different interface behavior for certain edge cases, will result in confusion.

To a certain extent, this applies not only to the visual interface design, but also the underlying software design. Software frameworks and APIs need to be designed in a way so they support engineers in effectively developing the required features in the specified terms of the code and design usability heuristics (Cazzola, W et al, 2005). In agile and iterative development environments, the evolutionary nature of both specifications and implementations represent additional challenges. In this regard, it proved critical to have specified principal usability heuristics, which guided designers and developers when conceptualizing the actual specifications for new or extended features.

The following paragraphs summarize the usability heuristics defined to improve the usability of the ASAP dashboard:

- *Single clicks within components should not reset the current search context.* This boils down to creating an interface which acts in a non-destructive way. Just clicking elements in the dashboard shouldn't trigger complex actions by themselves without any further explanation. Instead, single clicks should keep the current search context, but give the user options for further actions starting off from the current context. The actual implementation of this is to always provide a contextual menu after a single click and the extended tooltips with contextual menus are the result of these considerations. This menu doesn't directly affect the current state of the dashboard, it just offers additional metrics and menu options. So after a single click, the user still has the option to cancel this action by hiding the menu again, therefore keeping the dashboard's current state. This non-destructive approach fosters a user's ability to experiment and explore the application without triggering unexpected actions.
- *Do not "hijack" native interactions.* While some Web applications integrate their own functionality, for example when using a right mouse click, we decided against this: It affects the expected behavior in each user's or application's context and limits the usage of a browser's native functionality, e.g. using options to open a link in a new window or browser tab. Another reason to not use the right mouse click is that it represents a desktop-centric interaction and would require alternatives for usage via mobile or touch-based devices.
- *Avoid icon-only interface elements, use text instead.* While a very limited set of icons like a magnifying glass to represent a search query can be expected to be understood by all users, in most cases it is a challenging task to link specific icons to actions in highly specialized domains without additional explanations. While it is tempting to enrich the visual appeal of an application by adding custom icons, the significant overhead in terms of the design process often leads to limited improvements in the actual usability of the application. This led to the decision to primarily use descriptive, self-explanatory text as interface elements with optional additional icons.

Qualitative Validation

The development process for the ASAP Dashboard followed a series of iterative deployments, followed by feedback cycles after each deployment. This user-centered iterative process, shown in Figure 2, focuses on usability and can be applied to both the implementation process, as well as the application itself. It promotes usability throughout the whole development lifecycle (Matera, M et al, 2006).

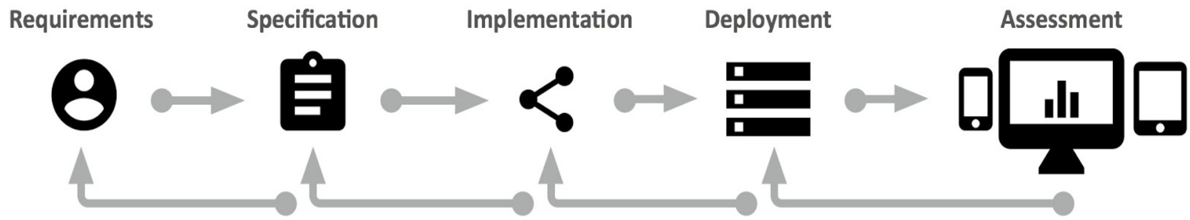

Figure 2. Diagram showing the development process, enables iterative qualitative validation

Overall, more than ten iterative version deployments and feedback loops were performed to integrate the new InfoViz services into the ASAP dashboard. The iterative process consists of the following steps:

- A *general requirements specification* based on the defined use cases builds the basis for further development. This initial stage includes high level discussions with WP leaders and user case partners, taking into account the defined scenarios of WP8 and WP9. The ability to store, manage and retrieve the structured datasets of the telecommunications use case, for example, was not sufficiently supported by the core platform, and had to be developed from scratch.
- Based on the general defined requirements, a team of designers and developers works on a detailed specification in terms of design and technology. This is an agile process and depending on the task, different methodologies like rapid prototyping or mockup creation can be involved. The outcome is a *detailed functional specification*. This work was supported by a self-hosted GitLab[4] instance, which features a flexible issue tracker and provides tools to foster asynchronous collaboration. In the case of the dashboard's statistical analysis features, the outcome of this stage were detailed mockups representing different application states, as well as a specification about the technical stack and required re-factorings.
- In a next step, the *implementation* of the functional specification is done. During this phase, designers and developers are in exchange to achieve the final desired result and overcome issues along the way. Again, GitLab was used in combination with other tools like SourceTree[5] to manage progress.
- An automated build process enables software *deployments* with little overhead and allows shorter feedback cycles.

[4] www.gitlab.com
[5] www.sourcetreeapp.com

- The deployed software is then assessed again by supervisors and stakeholders in terms of functionality and usability. Depending on the outcome of the assessment, this triggers either another round of this process or concludes the task.

System Usability Scale (SUS)

Once the developed application supported the specified use cases, 11 users attended a webinar (recruited among the use case partners as the main target group, complemented by other researchers not participating in the dashboard development efforts). After the introduction of the final version of the ASAP Dashboard, users explored the provided functions and filled out an online questionnaire including a series of open-ended questions as well as *System Usability Scale* (SUS) assessment to measure overall usability via a standard 10-item questionnaire with five response options - from *strongly agree* to *strongly disagree* (Brooke, 2013).

The received feedback from open-ended questions shows that users generally consider the ASAP Dashboard a comprehensive and easy to use system after proper training. The flexibility to combine multiple data sources with various options for further analysis was pointed out as a major strength of the platform and is in line with the goals set by the ASAP project. Still, the complexity of the dashboard and delivering too much information via multiple coordinated views can be overwhelming for first-time users with little prior experience in using complex analytics applications.

The average SUS obtained via the questionnaire is 74.5, which is a good result that exceeds the average score of 70 reported in the literature (Bangor et al., 2009).

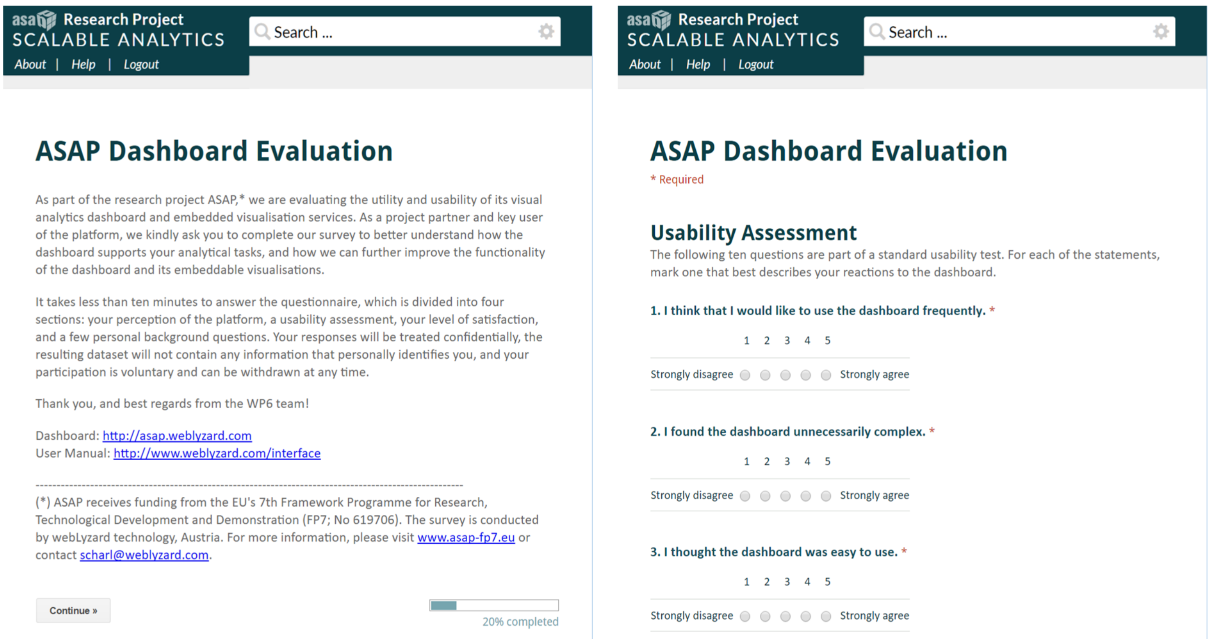

Figure 3. Screenshot of the online questionnaire to conduct the ASAP dashboard evaluation, including the standardized ten-point *System Usability Scale* (SUS) assessment

Dashboard Performance

During the final evaluation, the activity of ASAP Dashboard users was monitored to ensure its stability and performance. A total of 139 initial requests triggered a full load of the dashboard. 551 additional AJAX requests were triggered by users while using the Web-based *Single Page Application*. We also measured the performance of individual features. The *geographic map* of the ASAP Dashboard fulfills the sub-second goal of ASAP, aggregating the location metadata of more than two million tweets in about 700 milliseconds. The *statistical indicators* from the telecommunications use case of WP9 require about 500 milliseconds on average.

Application of Usability Heuristics

The following sections describe how the usability heuristics were applied to different components of the ASAP Dashboard, and include information on the results of the user questionnaire where applicable. To enable the integration of the ASAP project's use cases (WP8, WP9), several new and extended components were required to support the analysis custom statistical indicators – e.g., geo-tagged activity patterns in telecommunications data. From a usability perspective, the challenges were to extend existing functionality without compromising performance measures, and without an unnecessary increase in user interface complexity. The following sections discuss how these challenges were addressed for individual components.

Drill Down Sidebar

The selection sidebar on the left side of the dashboard originally was an organized list of defined topics and search associations. The requirements of ASAP triggered an evaluation and structural redesign of the component. A visual update of the sidebar was necessary to keep the dashboard interface consistent and easy to use. A cleaner, re-arranged layout supports the incorporation of additional settings and functionalities.

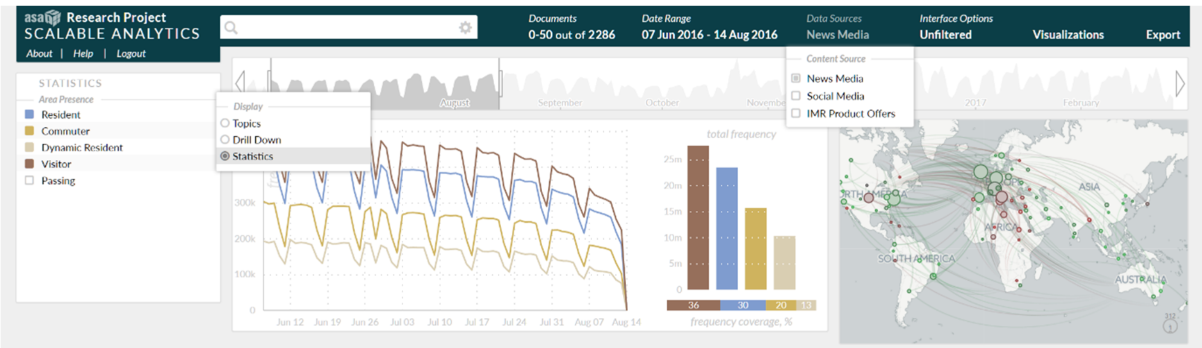

Figure 3. The upper section of the ASAP Dashboard with header, temporal controls and statistics sidebar showing selected indicators in the trend chart component (in this example, WIND area presence data from September to November 2016, classified by user type)

With these goals in mind and considering the ASAP WP9 use cases with a focus on analyzing telecommunications data, the content of the sidebar was split into multiple sections: *Topic Management*, *Metadata Drill-Down* and *Statistical Indicators*. The upper section contains the pre- and user-defined topics used as search bookmarks and grouped in categories. Each category has a title and is separated with a horizontal line from the category entries, similar to the header dropdown menus. The lower section consists of two parts: The current search term window and the list of associations.

The following visual tweaks were implemented to improve overall readability: (i) realignment of text and numbers, (ii) number formatting with a white space for the thousand separators, (iii) abbreviation for large values, and (iv) alternate white and gray backgrounds for all list items. Additionally, the categories can be collapsed or expanded according to use case requirements, hiding or revealing the contained items.

Each main section's title is coupled with a menu available via a so-called "*hamburger icon"* (three horizontal lines). The menus allow switching between different views:

- The *Topic Management* section is switchable to the *Metadata Drill Down* and the *Statistical Indicator* view.
- The *Associations* section is switchable to the *Search History* view.

This functionality replaces traditional tab elements. Other topic sidebar items (topic categories, topics, associations and search history entries) are coupled with a settings menu available on hover via a *gear icon*. Like the header dropdown menus or the adaptive tooltips, these menus can be extended with additional features without cluttering the main dashboard layout. An example is the display of *bar charts* that represent the currently indicated count values and allow their visual comparison at a glance.

The *Statistical Indicator* section shown in Figure 3 allows the analysis of custom statistical observations like telecommunications metadata. Its design and functionality are similar to the *Topic Management* view. Different attributes are grouped into categories, whose items are selectable for comparison in the trend chart. The similarity across sections not only minimizes the required training efforts, but also allows treating pre-annotated metadata in a similar fashion to other dashboard components.

Trend Chart

The chart window was reorganized and improved to support various new features, while preserving a clean look and feel. The expanding set of functions was separated into two groups: *global window settings* and *specific chart features*. Similar to the header drop-down menus, all specific chart features - dataset types, moving average options and chart switches - are put into the context menu to the left of the charts' window. The context menu, or "floating menu", is a dynamically expandable controller without that only becomes visible on hover. The global window settings remain available in the upper right corner in the form of icons. This allowed to reduce the complexity and visual clutter of the window and eliminate obsolete tab elements.

All charts were re-implemented using the D3.js framework (Bostock et al., 2011), removing other third-party libraries and dependencies. This approach enabled a more customized functionality, appearance and layout implementation.

Selected chart elements such as gridlines and labels were revisited and aligned with the new look and feel. The previous *pie chart* and *donut chart* components were replaced by a *bar chart* component. Additionally, data legends with topic information were added to the display when exporting a chart export. Figure 4 shows a screenshot of the final implementation.

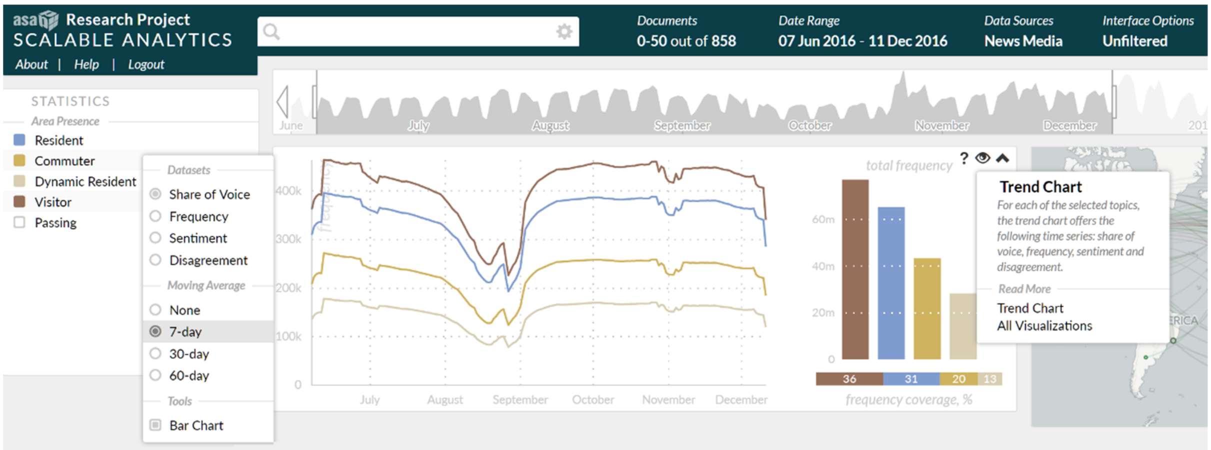

Figure 4. The trend chart component showing various statistical indicators including the *contextual menu* to the left as well as the *help* tooltip to the right.

Temporal Controls

Serving as an alternative for the calendar-based date range selector, the timeline-based date range selector is a full-scale visualization that displays the entire dataset in form of the area graph, where the vertical axis encodes the total frequency count and the horizontal axis encodes time. The actual selection of the desired range solely relies on the interactive functionalities of the module.

The window over the main area represents the currently selected range between two dates. It can be resized and/or dragged along the timeline to trigger a new selection and update the portal. The arrow buttons are used for moving the window forwards or backwards incrementally.

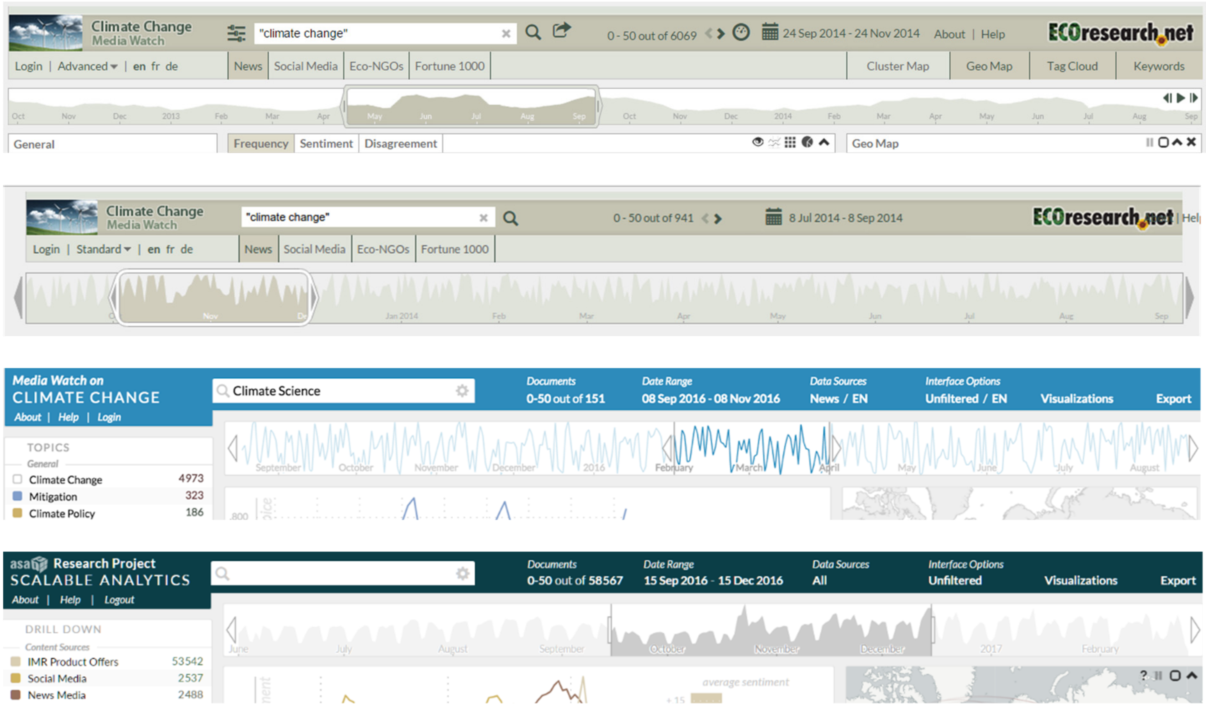

Figure 5. Design iterations I-IV for the temporal controls of the ASAP dashboard

The main features of the component had their appearance improved together with the other portal's modules. Initially, the timeline shared the color scheme with the portal's header to indicate its function as a global filter. The darker brown color was assigned to the selected chart area inside the window to create contrast with the deselected light green area. The buttons were originally placed in the upper right corner of the module (Figure 5 I).

When the new lightweight design was integrated, the timeline's appearance was updated accordingly (Figure 5 IV). Neutral colors were applied to its elements, such as dark gray for selected area, light gray for deselected area and medium gray for handlers, buttons and date labels. The window's border became visually lighter, emphasizing the selected part itself. The buttons moved to the sides of the area graph, making their use more intuitive while seamlessly combining with the window's left and right resize handlers. Taking advantage of the interactive dropdown menus, both calendar and timeline date range selectors were integrated into the menu available via the "Date Range" section of the header.

Geographic Map

To support the visualization of use case data, we investigated ways to visualize multiple indicators from various sources in a geospatial context. The requirement was that the geographic map developed in T6.2 should support the display of data gathered via the Statistical API in addition to metadata extracted from social media content.

Most reference implementations that we investigated had shortcomings in terms of both perception and aesthetics: For example, combining different types of layers (e.g. choropleth and point data; hexagon clustering and point data; using multiple icons for different types of data) distorts the represented data, makes comparing datasets more difficult if not impossible, and leads to visual clutter. The same is true for using small multiples of pie or bar charts within geospatial visualizations. Figure 6 shows a collection of references demonstrating these issues.

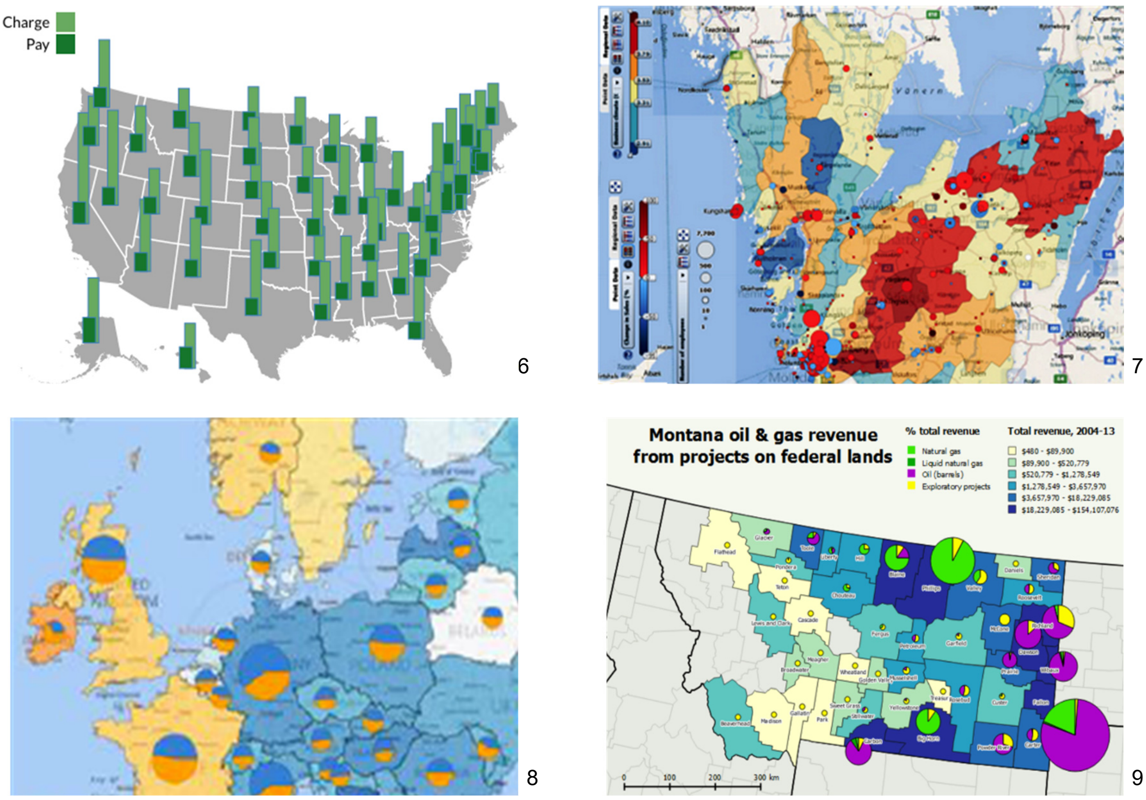

Figure 6. A collection of references demonstrating perceptional and aesthetic issues when comparing multiple datasets with different visual means within the same visualization

Given that the geographic map of the ASAP Dashboard is part of a set of tightly coupled data exploration components, the functionality had to be carefully considered to not overwhelm and confuse users. The chosen solution is a combination of server- and client-side geospatial clustering in combination with multi-colored circular markers

[6] https://vida.io/documents/s5qo5Gwrct5HNxAD2
[7] http://ncva.itn.liu.se/great-statistics-visualization/map-layers
[8] https://www.researchgate.net/figure/261456445_fig3_Figure-4-The-layer-architecture-of-the-choropleth-map-component
[9] http://www.pwypusa.org/workshop-2-calculating-and-visualizing-useiti-data-using-qgis/

that enables the adaptation of the display upon interactions. The dynamic clustering allows a drill down in real time from high-level views to regional and local data. The markers enable users to compare multiple datasets, like phone calls and the regional distribution of social media coverage. This approach can be generalized to visualize arbitrary metadata for different use cases.

We experimented with several ways to combine data overlay techniques and densify the information on display, keeping in mind usability and data perception requirements. In Figure 7, the geographic map shown on the right uses color to distinguish sources, and data point size for the data attribute. Our approach favors visual simplicity – but when combined with interaction and data highlighting techniques, it supports a range of advanced data exploration features. While in terms of implementation it would be straightforward to combine e.g. a choropleth base layer with additional layers using circles, such combinations tend to map data to different visual attributes (e.g. country shape, circle size, color) and lead to visual clutter, therefore slowing and skewing the perception of results.

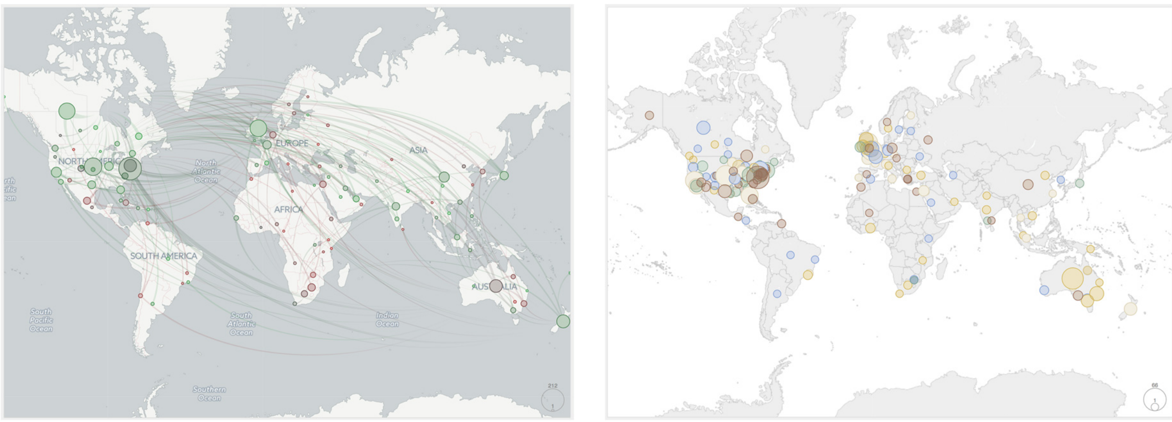

Figure 7. Comparison of two different geographic map modes: Sentiment distribution including trajectories linking source and target location (left); alternate color coding to compare multiple statistical indicators (right), using two different base layers.

Adaptive Tooltips and Context Menus

The ASAP dashboard introduced tooltips and contextual menus. In previous versions of the platform, users could click on elements like keywords or visualization elements and trigger a search, but these actions were hard-coded and sometimes non-intuitive. The actions also changed the search context of the dashboard's current view. The goal of the tooltip and contextual menu component was to (i) provide a standardized way to gain more insights into specific data points, (ii) allow the implementation of actions in context of a given component, and (iii) add non-destructive options for data exploration. The result is a flexible design and coding framework to create customized tooltips and contextual menus for synchronized dashboard components.

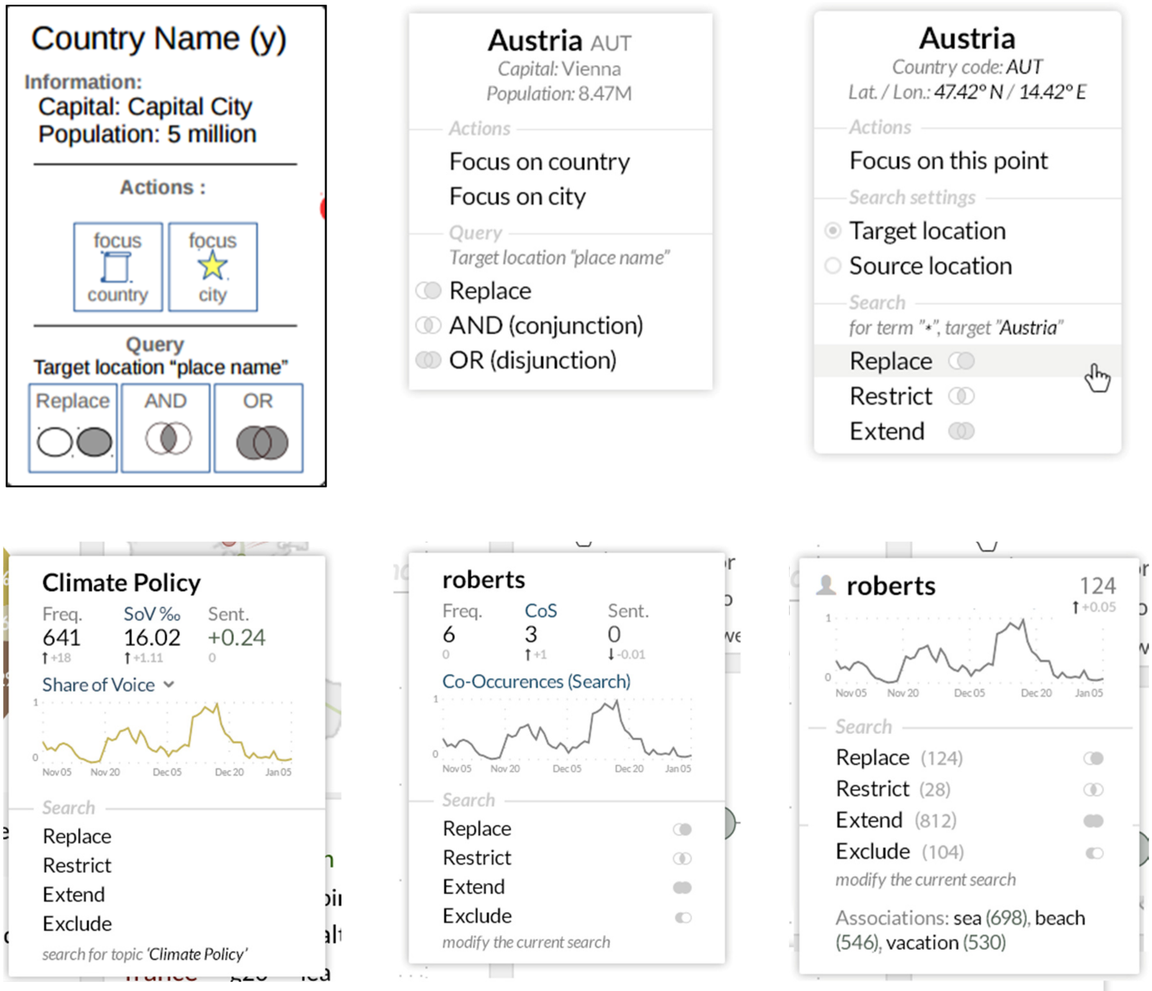

Figure 8. Design iterations of the first and second phase of tooltip implementation

The tooltips and contextual menus were designed and implemented in two phases. Initially, WP6 focused on establishing the general design principles and functionality (first row in Figure 8). In a second step, we improved capabilities and performance in terms of showing contextual information to a data point as well as providing additional functionality to the search options by adding small line charts, trending indicators and hovering options (second row in Figure 8).

The final version as shown in Figure 9 features a contextual line chart as well as trending indicators. This enables a detailed analysis of data points within the dashboard without triggering a new search – allowing the user to stay within the context of the current dashboard. The search options are enriched with previews of the number of matches that an actual search would trigger. Hovering the options dynamically updates the line chart giving the user the option to quickly compare the frequency distribution of each search option.

The tooltips and contextual menu of the ASAP dashboard were developed considering the high-performance requirements of the project. Based on a test corpus of one million news media articles, the loading time for the complete tooltip and contextual menu averages at 320 milliseconds.

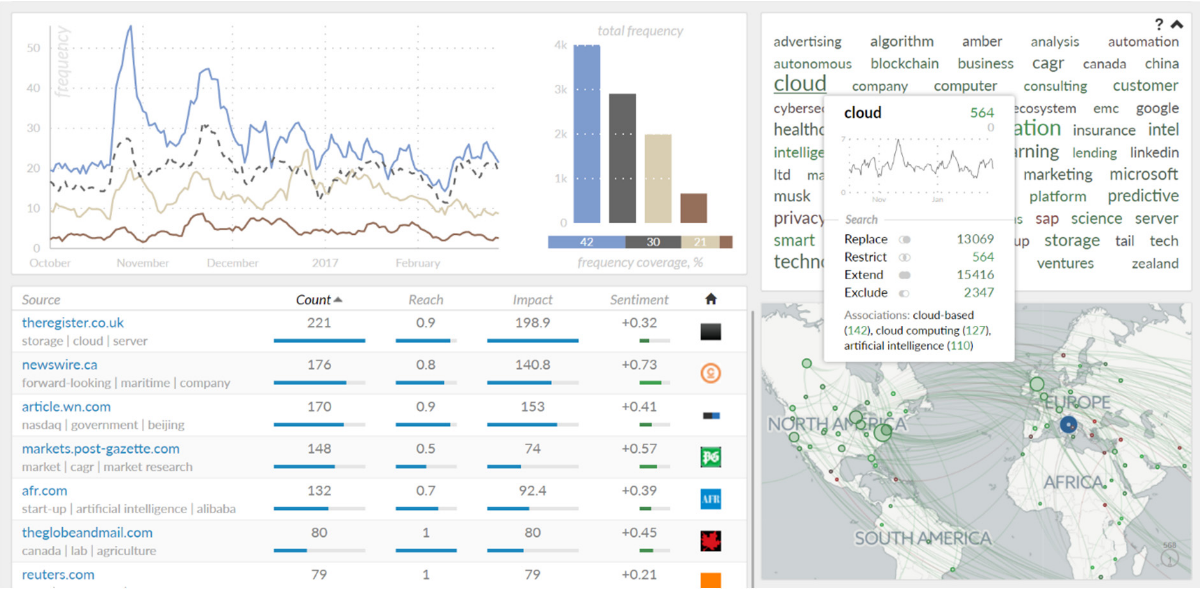

Figure 9. Screenshot of final tooltip version including contextual menu options

Summary

This deliverable D6.5 summarizes the ongoing usability evaluations conducted in T6.4 of the ASAP project. It describes the efforts to establish and validate usability of the ASAP Dashboard reported in D6.4 – focusing on describing the applied usability methods based on the expertise of participating researchers, as well as the feedback of users collected via an online questionnaire.

The deliverable demonstrates that the described workflow allowed designers and developers to incorporate usability considerations as a core component into their workflow, testing and improving new and revised components iteratively as part of an evolutionary development process.

A webinar presented the final dashboard version and collected user feedback through a series of open-ended questions and a standardized 10-item *System Usability Scale* (SUS) assessment, which was chosen over task-based quantitative measures to account for the dashboard's exploratory nature. An overall SUS score of 74.5 reflects a favorable perception of the interface. Individual comments by the participants mentioned the complexity of the ASAP dashboard, but pointed out that it is fast and easy to use after appropriate training, while offering a rich and comprehensive feature set.

References

Bangor, A., Kortum, P. and Miller, J. (2009). “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale”, *Journal of Usability Studies*, 4(3): 114-123.

Brooke, J. (2013). “SUS: A Retrospective”, *Journal of Usability Studies*, 8(2): 29-40.

Cazzola, W., Pini, S. and Ancona, M. (2005). The Role of Design Information in Software Evolution. *RAM-SE'05-ECOOP'05 Workshop on Reflection, AOP, and Meta-Data for Software Evolution.* W. Cazzola et al. Glasgow, UK: 59-72.

Fry, B. (2004). *Computational Information Design*, Boston: Massachusetts Institute of Technology. PhD Thesis.

Hubmann-Haidvogel, A., Scharl, A. and Weichselbraun, A. (2009). “Multiple Coordinated Views for Searching and Navigating Web Content Repositories”, *Information Sciences*, 179(12): 1813-1821.

Matera, M., Rizzo, F. and Carughi, G.T. (2006). "Web Usability: Principles and Evaluation Methods", *Web Engineering.* Eds. E. Mendes and N. Mosley. Berlin: Springer. 143-180.

Scharl, A., Weichselbraun, A., Göbel, M., Rafelsberger, W. and Kamolov, R. (2016). "Scalable Knowledge Extraction and Visualization for Web Intelligence", *49th Hawaii International Conference on System Sciences* (HICSS-2016). Kauai, USA: IEEE Press. 3749-3757. [Best Paper Award].

FP7 Project ASAP

Adaptable Scalable Analytics Platform

End of D6.5
Usability Report

WP 6 – Information Visualization
webLyzard technology

Nature: Report
Dissemination: Public