DELIVERABLE SUBMISSION SHEET

To: Susan Fraser *(Project Officer)*

EUROPEAN COMMISSION
Directorate-General Information Society and Media
EUFO 1165A
L-2920 Luxembourg

From:

Project acronym: PHEME Project number: 611233

Project manager: Kalina Bontcheva

Project coordinator The University of Sheffield (USFD)

The following deliverable:

Deliverable title: Usability Evaluation Report

Deliverable number: D5.3

Deliverable date: 30 December 2016

Partners responsible: MODUL University Vienna (MOD)

Status: ☒ Public ☐ Restricted ☐ Confidential

is now complete. ☒ It is available for your inspection.
☒ Relevant descriptive documents are attached.

The deliverable is:

☒ a document
☐ a Website (URL: ....)
☐ software (...........................)
☐ an event
☐ other (.......................)

Sent to Project Officer:	Sent to functional mail box:	On date:
Susan.Fraser@ec.europa.eu	*CNECT-ICT-*611233*@ec.europa.eu*	03 February 2017

FP7-ICT Strategic Targeted Research Project PHEME (No. 611233)

Computing Veracity Across Media, Languages, and Social Networks

D5.3 Usability Evaluation Report

Arno Scharl, Tobi Schäfer (MODUL University Vienna);
Anna Kolliakou (King's College London)

Abstract

FP7-ICT Strategic Targeted Research Project PHEME (No. 611233)
Deliverable D5.3 (WP 5)

This deliverable summarizes work conducted in WP5 of the PHEME project between September 2015 and January 2017. The goal of T5.5 is to conduct and document usability evaluation in regards to the PHEME Visual Analytics Dashboard developed in WP5, which is based on a multiple coordinated view approach to explore the veracity intelligence extracted by the content analytics methods from WP2, WP3 and WP4.

Keywords: visual analytics, veracity intelligence, validation, usability

Nature: Report (R)
Date of Delivery: 03 Feb 2017

Dissemination: PU
Reviewed By: Kalina Bontcheva

PHEME Consortium

This document is part of the PHEME research project (No. 611233), partially funded by the FP7-ICT Programme.

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP, UK
Tel: +44 114 222 1930
Fax: +44 114 222 1810
Contact person: Kalina Bontcheva
E-mail: k.Bontcheva@dcs.shef.ac.uk

Universität des Saarlandes
Language Technology Lab
Campus
D-66041 Saarbrücken
Germany
Contact person: Thierry Declerck
E-mail: declerck@dfki.de

MODUL University Vienna
Am Kahlenberg 1
1190 Wien
Austria
Contact person: Arno Scharl
E-mail: scharl@modul.ac.at

Ontotext AD
Polygraphia Office Center fl.4,
47A Tsarigradsko Shosse,
Sofia 1504, Bulgaria
Contact person: Georgi Georgiev
E-mail: georgiev@ontotext.com

ATOS Spain SA
Calle de Albarracin 25
28037 Madrid
Spain
Contact person: Tomás Pariente Lobo
E-mail: tomas.parientelobo@atos.net

King's College London
Strand
WC2R 2LS London
United Kingdom
Contact person: Robert Stewart
E-mail: robert.stewart@kcl.ac.uk

iHub Ltd.
NGONG, Road Bishop Magua
Building, 4th floor
00200 Nairobi
Kenya
Contact person: Rob Baker
E-mail: robbaker@ushahidi.com

SwissInfo.ch
Giacomettistrasse 3
3000 Bern
Switzerland
Contact person: Peter Schibli
E-mail: peter.schibli@swissinfo.ch

The University of Warwick
Kirby Corner Road
University House
CV4 8UW Coventry
United Kingdom
Contact person: Rob Procter
E-mail: rob.procter@warwick.ac.uk

Table of Contents

Executive Summary

The deliverable D5.3 summarizes the work conducted in Task T5.5 of the PHEME project. It includes documentation of the chosen software design process in terms of usability, and reports the results of the usability evaluation.

The deliverable describes the approach taken to make usability evaluation part of the core workflow to develop the PHEME Visual Dashboard (D5.2.2), focusing on the components that support the interactive exploration of veracity intelligence extracted by the content analytics methods from WP2, WP3, and WP4.

Following an iterative systems development approach, rapid feedback cycles and agile software development have been instrumental in the conceptualization and implementation of the PHEME dashboard. The deliverable describes usability improvements and evaluation for individual components by documenting design decisions and results from formative user tests.

Introduction

The overall goal of WP5 is to build visual analytics tools to interactively explore the veracity intelligence collected in WP6 of PHEME, including visualizations of geospatially and semantically referenced information across news media and social networks. T5.5 is about usability evaluations of the work done up to T5.4 which is about integrating the developed tools and providing a veracity intelligence dashboard configuration according to the specific requirements of a user.

The two use cases of WP7 and WP8 provided opportunities to assess and improve the interface. The dashboard enables gaining insight into popular issues that are being discussed related to the health domain (WP7), with a special focus on rumours and misconceptions, mental health and the pharmaceutical industry. The usability considerations and evaluations put special emphasis on the dashboard's dynamic rendering of metadata attributes along multiple semantic dimensions.

The deliverable D5.3 summarizes the work conducted in Task T5.5 of the PHEME project. This deliverable includes documentation of the chosen software design process in terms of usability and details the results of the usability evaluations.

Usability Evaluation Methodologies

To gain insight into the user experience of the PHEME dashboard following the initial deployment, usability evaluations were conducted. This deliverable D5.3 focuses on the different internal expert evaluations, while D7.3 takes external evaluations with more general users into account. The aim of these evaluations was to determine strengths and weaknesses of the interaction design and was done by using three types of assessments:

- *Usability inspections* investigated the interface design against recognised 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-centred design approach;
- *Formative usability testing,* observing users while working on predefined tasks in realistic settings.

The evaluations were done in iterative cycles depending on the project status: Following a user-centred 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, B. J., 2004). During the implementation phase, *usability inspections* and *qualitative validation* were integrated into the development process. Finally, *formative usability testing* was done: Users were observed while working on predefined tasks. Since the PHEME dashboard is a system aimed at experts, even despite usability efforts to increase ease of use, the application still requires some basic training. So in preparation for the user tests, a training session in the form of a screencast was created. This acted as an introduction to the PHEME dashboard and covered the description and demonstration of the following features necessary to complete the user tests: Search and filter functionality, document result list display options, data sources, visualization, topics and drill down features, associations, trend charts including optional features such as moving averages and tooltip features.

Feedback from earlier initiatives (e.g. the FP7 Project *DecarboNet)* or early adopters such as the *National Oceanic and 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. A large number of different components and can be daunting for new users without a technical background. So besides fulfilling the basic requirements in terms of functionality, special focus was put on improving ease of use and usability of the existing underlying platform including considerations to avoid additional complexity when implementing new features.

Usability Heuristics

The PHEME 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. Still, 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. On the other hand, navigating web sites is mainly done by hovering elements and using single mouse clicks. Browser applications themselves combine these paradigms. For the average user, these different contexts are not always easy to comprehend. So creating web based single-page applications differs not only in technological terms from classic web development, it is also a challenge in regards to considering various overlapping interface paradigms depending on the user's context.

Since the PHEME Dashboard offers a rich feature set and supports multiple visualisations within a single application view (Scharl et al., 2016), we had to carefully make decisions in regards to not complexifying the user experience. In order to keep a consistent interface, we defined specific usability heuristics for the dashboard. To describe the reasoning and importance of this, let's compare it to laws from physical theory: The law of gravity describes a certain behavior we can witness in everyday life and we take more or less given. For example, outdoors here on Earth, if we pick up a stone and drop it again, we expect it to fall back on the ground. However, if – for whatever unknown reason – the stone would keep floating in mid air, we would be most certainly confused and irritated, and our reaction to the unexpected behaviour could be curiosity or anxiety. The same applies to software interfaces: Inconsistent usage of interface paradigms or usage patterns increase complexity and irritate users. While some of the following usability heuristics might seems trivial at first glance, the actual challenge for a software designer is to stick to them throughout the design and implementation phase. For the user, providing for example a different interface behavior for certain edge cases, will feel like witnessing a rock floating in mid air. To some regard the same applies not only to the visual interface design, but also the underlying software design. The software's underlying 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). Additionally, in agile and iterative development environments, the evolutionary nature of both specifications and implementations lead to even more challenges. In this regard it proved critical to have specified principal usability heuristics which can act as fallback helpers when conceptualizing additional actual specifications for feature extensions.

The usability heuristics we defined to improve usability are the following:

- *Single clicks within components shouldn't reset the user's 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, the behavior of single clicks should be to 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. This menu doesn't directly affect the current state of the dashboard, it just offers some 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.

- *Don't hijack the host's application specific 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 a given user's or application's context and limits the usage of a browser's native functionality, e.g. using options to open a link either in a new window or browser tab. Another reason to not use the right mouse-click is that it's a desktop-centric interaction and would require alternatives for usage via mobile or touch based devices anyway.

- *Avoid icon-only interface elements, use text instead.* While a very limited set of icons like a magnifying glass to represent a search action can be expected to be understood by users, in most cases it is a very challenging task to establish specific icons for actions in highly specialized domains without additional explanations. Custom icon design adds significant overhead to the design process too. So while it's tempting to enrich the visual appeal of an application by adding custom icons, another drawback is that this adds significant overhead in terms of required human and time resources to the design process with limited improvements to the actual usability for 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 PHEME dashboard was done using iterative deployments. The development methodologies allowed to include feedback cycles after each deployment. This iterative process, shown in Figure 1, involves considering usability at its core and can be applied to both the implementation process, as well as the application itself. This iterative approach promotes usability throughout the whole development lifecycle (Matera, M et al, 2006).

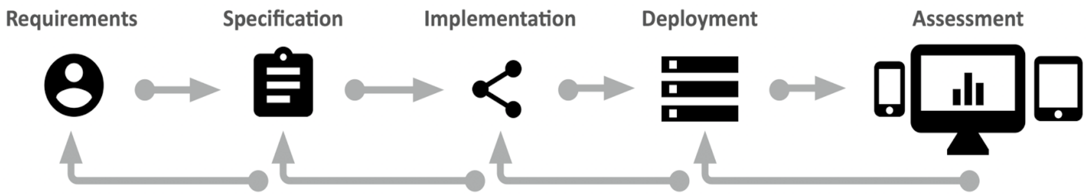


Figure 1. Diagram showing the development process which enables iterative qualitative validation.

Overall, more than 20 iterative version deployments and feedback loops were done for the PHEME dashboard to develop the rumour and stance analysis features. 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 stakeholders and project managers and took into account the defined use cases of WP7. For example, PHEME's use case about rumour detection was not sufficiently supported by the underlying existing technology and had to be taken into account.

- 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 is documented using the software GitLab which features a flexible issue tracker. The software's approach is to provide tools which foster asynchronous working environments. In the case of the dashboard's rumour and stance 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 refactorings.

- After this, *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 to manage progress.
- An automated build process enables software *deployments* with little overhead and allows shorter feedback cycles.
- The deployed software is then assessed again by supervisors and other 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.

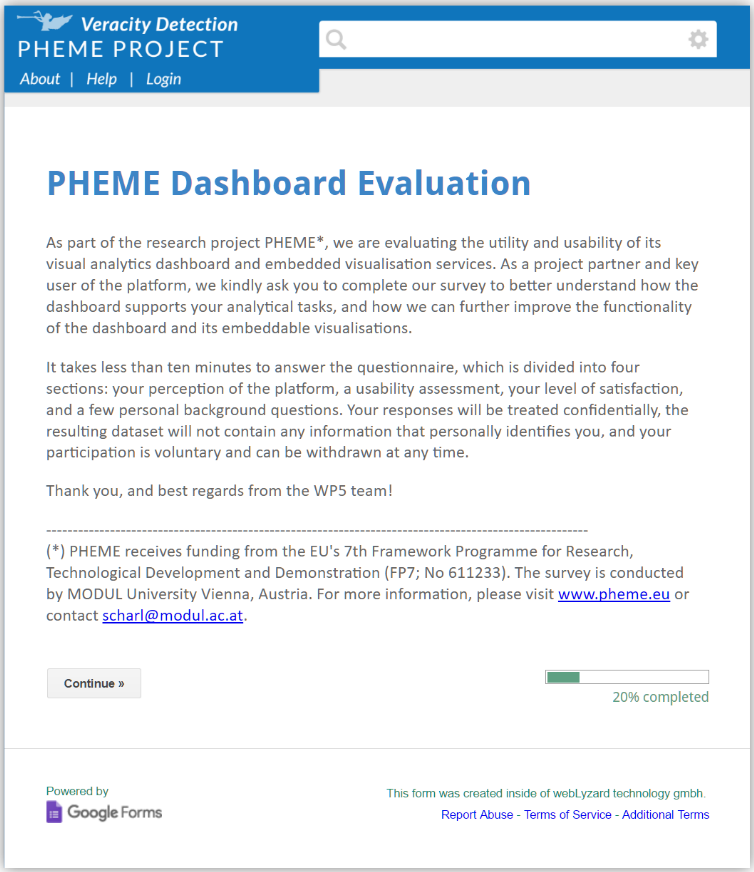
Formative Usability Testing

Once the developed application supported the required use cases, formative usability testing was done. Six users were given a list of 19 tasks which they were asked to complete, in one sitting, using the PHEME dashboard. Users were also provided with a two-sided A4 summary sheet offering an overview of the main dashboard sections and features for reference, if needed. For each task, they were asked to indicate the correct answer, if applicable, and given the option to also record their thoughts and comments. Each task was, then, assigned a score of 0, 1 or 2 corresponding to non-completion, completion with help or difficulty and easy completion, respectively.

The tasks ranged widely and explored a number of functions from general interface usage such as sources and configuration to trend charts and visual analytics. A number of tasks more specific to WP7 requirements such as sentiment and rumour analysis, was also tested.

Overall, users on average could solve 79% of the given tasks and 67% could be completed easily. Participants were able to complete the test in a time range from 34 to 45 minutes. Participants were also asked to provide informal feedback be means of a questionnaire implemented via Google Forms (see Figure 2). The questionnaire contained a standard evaluation following the *System Usability Scale* (SUS) format (Brooke, 2013; Sauro, 2011); the results of the corresponding ten questions will be reported in D7.3.

Test users overall impression of the PHEME Visual Dashboard was positive in terms of the rich feature set for media and natural language analysis including exploration of emerigng PHEMEs. While some users were positive about the ease of use and the intuitive and fast interface, negative feedback included that advanced features require initial training and are not without their complexities. Looking at the background of participants, it seems users with a better general understanding of more advanced web applications found the dashboard easier to use than others.


Veracity Detection
PHEME PROJECT
About | Help | Login

PHEME Dashboard Evaluation

As part of the research project PHEME*, we are evaluating the utility and usability of its visual analytics dashboard and embedded visualisation services. As a project partner and key user of the platform, we kindly ask you to complete our survey to better understand how the dashboard supports your analytical tasks, and how we can further improve the functionality of the dashboard and its embeddable visualisations.

It takes less than ten minutes to answer the questionnaire, which is divided into four sections: your perception of the platform, a usability assessment, your level of satisfaction, and a few personal background questions. Your responses will be treated confidentially, the resulting dataset will not contain any information that personally identifies you, and your participation is voluntary and can be withdrawn at any time.

Thank you, and best regards from the WP5 team!

(*) PHEME receives funding from the EU's 7th Framework Programme for Research, Technological Development and Demonstration (FP7; No 611233). The survey is conducted by MODUL University Vienna, Austria. For more information, please visit www.pheme.eu or contact scharl@modul.ac.at.

Continue »

20% completed

Powered by Google Forms

This form was created inside of webLyzard technology gmbh.
Report Abuse - Terms of Service - Additional Terms

Figure 2. Questionnaire for the evaluation of the PHEME Dashboard

During the time period the user tests were conducted, the PHEME Visual Dashboard's activity has being monitored to ensure it's stability and performance. In the testing period in tota 99 initial requests were done, meaning the PHEME Dashboard was fully loaded this number of times. 3679 additional ajax requests were triggered by users while using the fully loaded single page web application. We also measured the performance of individual features. General search functionality and components like the geographc map which visualize its data without the need for additional data aggregations and transformations average at 635ms. More complex visualizations like the Social Network Analysis visualization take 2 to 10 seconds based on the size and detail of the required dataset.

Usability Evaluation

The following sections describe how the usability heuristics were applied to different components of the PHEME dashboard and include the outcome of the formative usability tests where applicable.

Search and Filtering Interface

The header of the dashboard represents the main interface component used for searching and filtering queries with custom settings and sets up the context for further analysis including rumour medata annotations. Within the PHEME project, the component has undergone a major redesign with the following preconditions and considerations in mind: the functionalities in the previously existing component grew over time and became more complex. Considering the requirements of future features, it became clear that the header needed a complete overhaul because incremental feature additions would only increase the complexity. The main design considerations for the updated component were to present the global configuration UI elements in a straightforward and self-explanatory way and consolidate different types of visual elements (icons, buttons, text, sidebar) into two consistent types - text buttons and context menus.

The re-design groups similar header functionalities in category sections which reveal additional settings via dropdown menus when hovered. These redesigned section features enabled the use of additional descriptive texts which made the header elements more self-explanatory. A category section consists of two text captions: the upper caption shows the section title and the lower caption displays the selected setting(s). In contrast to the formerly used tabs shown within the header component, the drop-down menus don't suffer from the same space restrictions and can be expanded vertically in line with the amount of available options. The dynamic behaviour of the header menu items was developed by evaluating different visual representations of the currently selected state in the dropdown menus. The grouping of the header options allowed to remove the multi-row layout, significantly reducing the complexity of the structure and decreasing the overall size of the header.

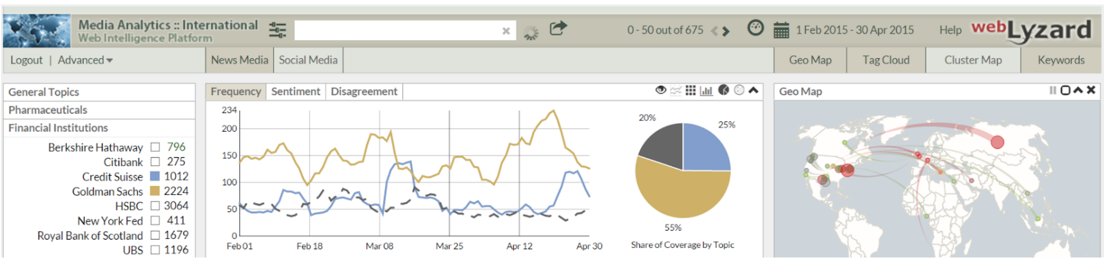


Figure 3. The initial header layout in 2015, prior to developing the PHEME Visual Dashboard

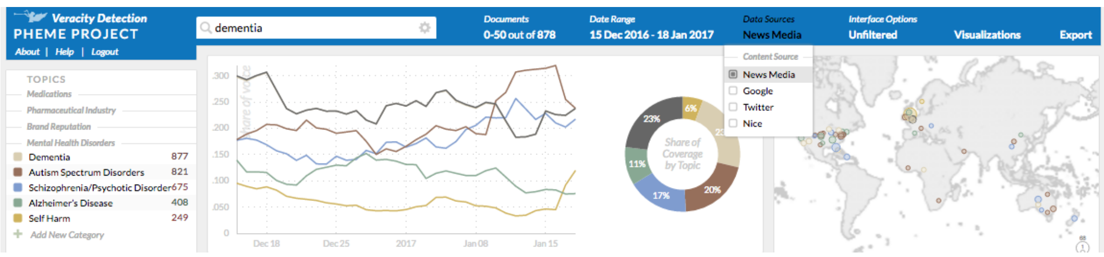


Figure 4. Screenshot of the redesigned header layout as of January 2017

One of the main global filtering elements in the dashboard - the search bar - became bigger and therefore more prominent, emphasizing its importance. The header component doesn't rely on customized, hand-crafted and oversized bitmap images anymore. Instead, the title element in the upper left corner is now configurable with custom two-row texts and an additional, optional SVG logo element that reinforces the dashboard's identity. The background color of the header is customizable too. A vivid color visually separates the header from the rest of the dashboard's modules, highlighting its functionality, and provides more contrast to facilitate the reading of the text captions.

The implementation of the revamped header is based on the RactiveJS framework. In contrast to the previous technique using a combination of plain JavaScript and jQuery, RactiveJS offers a more concise and structured way of dealing with the application states, user events and DOM manipulation. The new interface concept turned out to be very flexible and easily extendable. Instead of cluttering the header's layout itself with additional elements, the dropdown menus can be used for implementing additional features, such as multi-lingual filters.

User tests showed that most tasks which involved the header's search and filtering functionality could be completed easily. Some users had difficulties with tasks which included additional combined usage of the lower left current search though.

Metadata Integration

In addition to the graph-based visualizations using PHEME's *Graphyte* library[1] outlined in a later section of this document, existing components have been extended to enable the integration and analysis of PHEME-specific metadata (e.g., *veracity* and *stance* information as well as pre-annotated *clustering*). From a usability perspective, the challenges were to both extend the existing functionaly without compromising existing use cases and avoid introducing a more complex interface. The following sections describe how we addressed this for individual components.

Drill-Down Sidebar

The selection sidebar, available at the left side of the dashboard, originally was an organized list of defined topics and current search associations. It was extended in the ASAP project to support additional views. This was used as a basis to implement additional extensions in regards to the drill-down sidebar to support the use cases of the PHEME project with a focus on rumour and stance analysis. This drill-down section can be enabled using the menu icon (three horizontal lines).

The Drill-Down section is the main component which allows the analysis of custom pre-annotated metadata like veracity or stance information. After switching the upper section

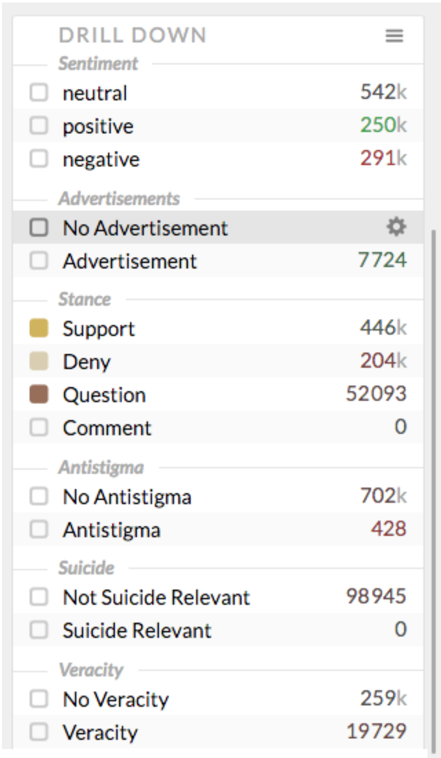


Figure 5. The search drill down menu including newly introduced metadata attributes

[1] www.github.com/weblyzard/graphyte

to the Drill-Down view, the metadata attributes of the current search term become available. Its design and functionality are similar to the Topics view: different attributes grouped in categories, items are selectable for comparison in the trend chart. Since the drill-down component is similar in appearance and functionality to the topic section, (i) user's are more easily able to learn how it works and (ii) while offering an interface for different use cases than the topic sidebar, its functionality allows to treat pre-annotated metadata in a similar fashion in combination with other components of the dashboard.

This component played a role in 8 of the 19 tasks from the formative usability testing. Tasks which included selecting predefined topics as well as configuring the trend charts using the sidebar could be solved easily. However, some users encountered difficulties completing very specific tests when using the drill-down sidebar.

Trend Charts

The window containing the charts was reorganized and improved in order to support the various new features and preserve the tidy look and feel. The expanding functionality set was separated in two groups: global window settings and specific chart features. Similar to the header drop-down menus, all specific chart features 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 jeopardizing the dashboard's interface and becomes visible when the window is hovered by moving the mouse pointer over it. The global window settings remain available in the upper right corner in form of icons. This allowed to reduce the complexity and visual clutter of the window and get rid of the obsolete tab elements.

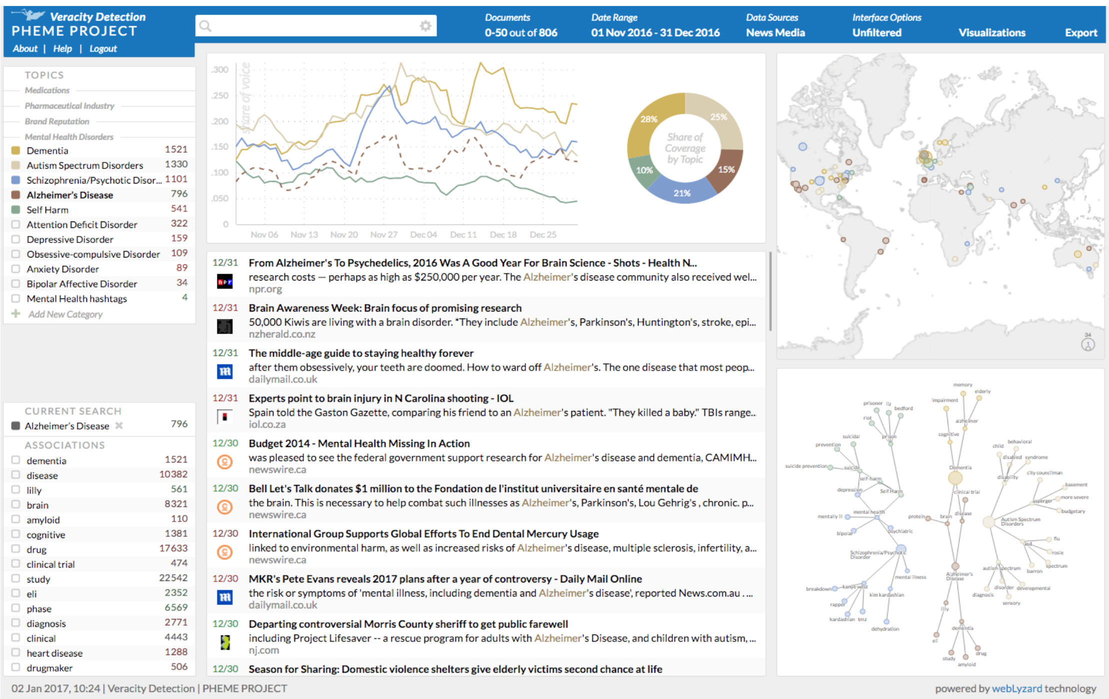


Figure 6. Screenshot of the full PHEME Dashboard with online coverage about *Alzheimer's Disease* between November and December 2016

The trend chart component itself was already agnostic to the type of data provided (e.g. if the data was based on topic or metadata filtering). This means the component itself needed only minimal adaptions from a technological viewpoint to support the additional metadata specific to the PHEME project. From a usability perspective this had it's advantages too: The interface and visual display is the same, just the type of data selected in the selection sidebar changes.

In formative tests the trend charts played a role in 4 of all 19 tasks (Questions 7, 8, 13 and 14). Users were able to solve 87% of these tasks. 79% of the tasks were completed easily.

Document View

The tab structure of the content view was transformed into the context menu. Some of the view-specific options, such as sorting options for the Documents/Sentences and level of detail for the Source Table/Source Map, migrated to the menu as well, reducing the visual load of the windows. The content view supports to switch to a full text version of indivdual documents. This view was extended to display additional custom metadata information including veracity scores.

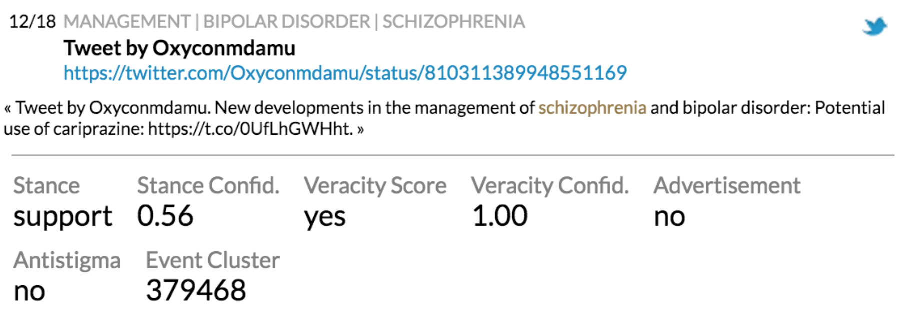


Figure 7. Detailed document view of a tweet including metadata

Source List

Additions to the source list include options to display aggregation on pre-annotated event clusters as well as the bar charts that encode the value fields for each entry. Located below the numbers, the visual indicators allow to quickly compare and identify the predominant values. Moreover, the readability of the list was improved by adding a gray background to every even row and line separators between the rows. All elements of the list were properly aligned, making the layout more organized.

The formative user tests included a task on how to use the source list (Task 5). All users were able to complete task, with two having some difficulties. The task included locating and enabling the source list view, sorting by specific indicators and identifying top sources.

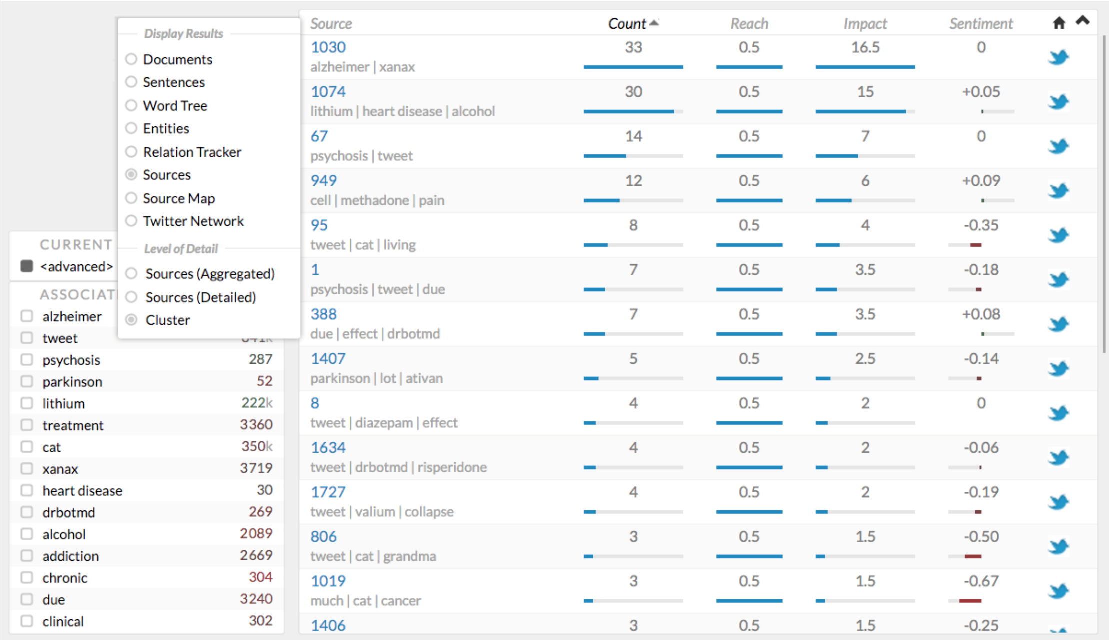


Figure 8. Source list with the newly introduced capability to aggregate results by cluster annotation

Source Map

Besides simply translating the source list into visual form, the source map version used in the PHEME Visual Dashboard is capable of visualising the pre-annotated event clusters. In order to do that more efficiently and intuitively, the appearance and the layout of the source map was improved. The look of markers was consolidated with other circle-based visualizations of the dashboard, the label and marker positioning are now executed more precisely, the grid was updated with standardized formatting of numbers and ticks.

Graph Visualizations

Cluster Map

Several visual tweaks were introduced to the cluster map to improve its look and feel. The cluster hull color was changed from orange to gray to keep the sentiment-based color scheme of the document nodes in focus and avoid skewing color perception of the sentiment values. The appearance of nodes was consolidated with other circle-based visualizations of the dashboard. A white outline was added to the cluster labels to improve readability. Additionally, the font size of the labels is adjusted dynamically depending on the dynamic zoom level of the visualization. Additional features and improvements were implemented which are described in more detail in D5.2.2 in the respective section on the cluster map.

Keyword Graph

After introducing support for multiple root nodes, the layout of the keyword graph was reconsidered. The positioning of the root nodes is arranged in a circular manner, while the satellite nodes are placed using techniques from the *Graphyte* library. The edge lengths are dynamically calculated by the *Graphyte* algorithms as well. To preserve consistency, the nodes' appearance was updated to match the design of the circle-based visualizations in the dashboard. The keyword graph was included in one task of the formative usability evaluations (Task 6). While four users had no issues with the visualization itself and were still able to complete the task, there was some feedback saying that the additional configuration options were confusing. It turned out this was due to that these interface elements didn't completely follow the defined usability heuristics, for example icons without further explanations were used as buttons. Like the cluster map, the feature improvements were already described in more detail in D5.2.2.

Social Network Analysis

The SNA graph is based on the keyword graph's implementation with addition of directional edges and network-like connections. Considering the varying size of nodes, the layout of this visualization model was optimized using Graphyte's algorithms in conjunction with collision detection. The look of the nodes was improved by applying the circle styling from other dashboard's visualizations. The various global display settings, such as showing sentiment/user image in the nodes and the centrality measures for the node sizes, were inserted into the floating menu.

Geographic Map

Similar to how other components of the dashboard work, the geographic map had to be extended to support the use cases to investigate rumour and stance information. The specific implementation of the selection sidebar enabled the same frictionless user experience to investigate the geospatial distribution of metadata like it does for the trend chart in regards to distribution over time. While previous versions of the geographic map supported focusing on a single search and its sentiment distribution only, the extensions developed for the PHEME project allow both the visualization of geospatical distribution across several predefined topics as well as custom metadata like type of source (e.g. social media authors, news media outlets, corporate web sites), veracity or stance without introducing a more complex interface.

The formative usability tests included one task about the geographic map (Question 9). Five out of six users were easily able to solve the task which included identifying the most prominent mental health disorder on a specific location.

Summary

This deliverable D5.3 summarizes the evaluation conducted in T5.5 of the PHEME project. It describes ongoing efforts to establish and validate usability of the visual dashboard reported in D5.2.2 – focusing on the expertise of participating researchers, the feedback of project partners and professional users collected via an online questionnaire, and the completion of predefined tasks by test users as part of experiments conducted in WP7. 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 as part of an evolutionary development process. Specific emphasis has been placed on additional metadata attributes provided by WP2 of PHEME, including automatically extracted per-document *Stance* and *Veracity* values. *Formative tests* showed that participants were able to successfully complete the majority of tasks. Follow-up projects will continue the evolutionary development process, both in terms of improving the core components as well as developing new visual tools based on PHEME'S *Graphyte* visualization library.

References

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

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.

Sauro, J. (2011). *A Practical Guide to the System Usability Scale: Background, Benchmarks, and Best Practices*. Denver, USA: Measuring Usability LLC.

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].

Matera, M., Rizzo, F., & Carughi, G. T. (2006). Web usability: Principles and evaluation methods. In *Web Engineering* (pp. 143–180). http://doi.org/10.1007/3-540-28218-1_5

Cazzola, W., Pini, S., & Ancona, M. (2005). The Role of Design Information in Software Evolution. In W. Cazzola, S. Chiba, G. Saake, & T. Tourwé (Eds.), *RAM-SE'05-ECOOP'05 Workshop on Reflection, AOP, and Meta-Data for Software Evolution, Proceedings, Glasgow, UK, July 15, 2005* (pp. 59–72). Fakultät für Informatik, Universität Magdeburg.

Fry, B. J. (2004). *Computational Information Design*. PhD Thesis, Massachusetts Institute of Technology. www.benfry.com/phd/dissertation-110323c.pdf