An Embeddable Dashboard for Widget-Based Visual Analytics on Scientific Communities

Michael Derntl  
RWTH Aachen University  
Advanced Community Information Systems (ACIS)  
Aachen, Germany  
derntl@dbis.rwth-aachen.de  

Stephan Erdtmann  
RWTH Aachen University  
Advanced Community Information Systems (ACIS)  
Aachen, Germany  
erdtmann@dbis.rwth-aachen.de  

Ralf Klamma  
RWTH Aachen University  
Advanced Community Information Systems (ACIS)  
Aachen, Germany  
klamma@dbis.rwth-aachen.de  

ABSTRACT

The web hosts massive amounts of data about scientific communities, for instance in bibliographic databases, research project databases, and more recently on the social web. These data sets provide key information on the activities and artifacts relevant to the community. Due to the scattered nature and sheer size of Web 2.0 data sets it is currently difficult for stakeholders to comprehensively and effortlessly monitor key indicators hidden in that data. To address this issue, we developed an embeddable, personalizable dashboard, which can be populated with widgets that visualize the results of arbitrary queries over relational databases in real time. As a pilot case the dashboard was deployed at LearningFrontiers.eu, a service and support portal for the European Technology Enhanced Learning community, to provide access to pre-configured visualizations of queries over their data sources. Community stakeholders evaluated the dashboard and generally responded positively in regard to its usability and usefulness. From a technical perspective the main advantage of the proposed dashboard infrastructure is the simple and seamless integration with a community’s existing data sources and web portals.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentations]: User Interfaces; H.2.8 [Database Management]: Database Applications; H.3.5 [Information Storage and Retrieval]: Online Information Services

General Terms
Design, Human Factors

1. INTRODUCTION

Scientific results are always socially created in so-called scientific communities [11]. It is quite obvious that quality of scientific results correlates heavily with the success of scientific communities. Consequently, there is a huge interest in finding out about success factors for scientific communities with respect to different stakeholder groups like funding agencies, policy makers, the public audience and the communities themselves. Most methods for measuring success—e.g. citation analysis or productivity indices—focus on the publications as the major output of scientific communities [5]. However, these classical success measures are difficult to apply to new, emerging communities. By the increase of third party funding, globalization in research and the establishment of new communication and collaboration means like the Web 2.0, scientific results have become even more scattered and available in many traditional and new media in different forms of abstraction. Applications like blogs, wikis or social networks enable creation and sharing of information for nearly every web user. This has spawned a vast and continuously growing source of information, complementing the information provided by traditional media. With current web technologies, information updates (e.g. twitter streams) can be presented also in a near-live experience to users. New research instruments, like networks of excellence were created by the EU research funding agencies to overcome fragmentation in certain areas and to drive excellence on a European level. In the area of technology enhanced learning four networks of excellence have been established so far, PROLEARN, KALEIDOSCOPE, STELLAR and GALA. All these networks have taken measures to collect and display valuable research & development information in various places, e.g. in deliverables [19], Mediabases [10], or concept maps [7]. Now, it is time to integrate and distribute the visualization process to support scientific community success by raising awareness and reflection from within the community and for other relevant stakeholder groups.

Data visualization tends to make the presentation of information simpler and more efficient. In particular, “visual representations and interaction techniques take advantage of the human eye’s broad bandwidth pathway into the mind to allow users to see, explore, and understand large amounts of information at once” [17]. Information visualization thus provides an intuitive method of acquiring and assimilating information. A nowadays more and more popular application of data visualization techniques are dashboards. The term “dashboard”, in reference to the control panel in front of the driver of a car, refers to a visual display of the most important information needed to achieve one or more objec-
tives, consolidated and arranged on a single screen so the information can be monitored at a glance [3]. A dashboard reduces a large amount of information to those parts in which the viewer is interested, and visualizes this information in a way that helps the viewer understand it quickly.

In regard to large and scattered community data sources, dashboards are a viable solution to present the contained information to the relevant stakeholder groups using visualizations of queries over the available data. This will support a community’s stakeholders in perceiving, understanding and reasoning about complex data, which is one of the main goals of visual analytics [12]. This paper presents an attempt to enable this by synthesizing existing web technologies, visualization APIs, and database technologies into an embeddable dashboard infrastructure.

The paper is structured as follows. In Section 2 we discuss dashboards and their use for visual analytics. In Section 3 we outline the requirements, architecture and implementation of an embeddable, widget-based dashboard. A pilot application of this dashboard in a service portal of the Technology Enhanced Learning community is presented in Section 4. In Section 5 we present an evaluation study of this pilot application, and Section 6 concludes the paper.

2. VISUAL ANALYTICS DASHBOARDS

In cars the dashboard instruments in front of the driver—e.g. speedometer, mileage indicator and fuel gauge—display the information the driver needs to safely operate the car. As she has to concentrate on the surrounding traffic, she cannot afford to scan the dashboard for lengthy time periods to find the required information chunks (e.g. observing the speed while entering the city limits). For this reason the dashboard has to convey the information at one glance, and each instrument has to be easy to read and interpret.

These requirements of an instrument panel in a car can also be applied to dashboards in the context of human-computer user interfaces that expose the users to massive amounts of data that may include unwanted noise or details. Turning information overload into an opportunity is the driving vision of visual analytics [8], and dashboards have played a central role in this regard. In general, the analysis of large amounts of data will be a difficult task without appropriate data aggregation and presentation. For instance, the detection of outliers or patterns within a data set will be difficult when scanning raw data. Dashboards simplify this task by aggregating, filtering, and presenting the data. A well-known example is the Google Analytics [http://www.google.com/analytics](http://www.google.com/analytics) dashboard, which presents key site traffic data to website operators.

Few [4] characterizes a dashboard as a visual display that represents the information needed to achieve specific objectives. This implies a reduction of the whole data set to just the relevant information which will then be visualized. The visualization itself also illustrates just the essential bits of the data while discarding all unnecessary information like long descriptions. Furthermore, to make the information accessible at a glance the dashboard should fit on a single screen [6]. This also leads to the challenge of visual dashboard design “to optimally use the available screen space for conveying as much information as possible to the user” [9].

A dashboard consists of a small number of components, each visualizing a relevant piece of information. The components should be visually separated and consistently formatted to improve the usability of the user interface [15]. As an extension, a dashboard can also contain interactive components enabling users to take a closer look on any displayed information [17], e.g. by showing more detailed information when the user hovers the mouse pointer over a chart region or by offering zooming, panning and filtering features. Such features are required to support users in their exploratory analyses which are central to visual analytics applications [3]. To adapt the dashboard to the user’s preferences, customization options can be provided, allowing the user to choose the dashboard components to display [4].

Dashboards have been used in a broad range of applications, for instance embedded in software (e.g. network monitoring) or as a component of a computer desktop (e.g. the Mac OS X dashboard). Nowadays, dashboards are typically realized using web widgets, i.e. small applications that are embedded in web sites and executed in web browsers, which makes them platform independent and usable without installation. Web widgets became popular with the rise of customizable browser start pages like Netvibes or iGoogle.

While Google, for instance, offers features for rendering OpenSocial widgets in web containers, a drawback of the current state-of-the-art is that there is no lightweight framework that enables the embedding of a complete dashboard infrastructure into arbitrary web sites operating on arbitrary data sources. A web-based, lightweight infrastructure to remedy this drawback is presented in the following section.

3. EMBEDDABLE DASHBOARD INFRASTRUCTURE

3.1 Requirements

The main objective of the work reported in this paper was to provide an embeddable, personalizable dashboard that allows to embed widget-based visualizations of arbitrary data base query results. This requires the conception of a web-based dashboard frame in which visualization widgets can easily be added, rearranged, or removed. Users can create new visualization widgets for other users to embed into their personal dashboards. A user management and widget store infrastructure is required to store user preferences between sessions and the available widgets, respectively.

Visually, a widget-based dashboard interface typically consists of (1) a dashboard frame, i.e. the container that hosts the widgets, and (2) a control pane which enables the user to search and browse for widgets and embed widgets. To enhance the usability, customization options are offered like adding/removing columns, width/height adjustment and re-ordering of widgets using drag-and-drop operations.

In our dashboard, the control pane must enable the end-user to browse the widget store for widgets that he or she can add to the personal dashboard view. To ease the discovery of available widgets the widget store can be searched by using either keyword search or filtering by category. Users should be enabled to generate a new widget by defining an SQL query statement for a particular data source as well as a visualization style for the query result. After the creation the user can store the newly created widget along with some metadata to the dashboard’s widget store. It should be possible to hide the control pane or detach the dashboard into a separate window to make optimal use of the available screen space for the visualization widgets.
3.2 Use Cases

The core functionality of the dashboard is presented in the use case diagram in Figure 1. There are three actors interacting with the dashboard system:

- **Guest**: guest users are visitors who are not authenticated in the current session. These users can interact with a pre-defined set of widgets that are part of the default dashboard layout, but these users cannot personalize the dashboard. Guests can log in or register to get access to additional functionality.

- **Authenticated User**: an authenticated user is granted access to personalization features, allowing her/him to add widgets or remove widget from her/his personal dashboard view. Besides that, this user also can rearrange or resize the dashboard components using drag-and-drop. These personalization actions are stored in the user preferences.

- **Widget Creator User**: this is a specialization of an authenticated user, who can also create new widgets by using the Widget Creator Application. New visualization widgets can be defined by selecting a configured database connection, specifying an SQL query on that database, as well as selecting visualization specific configuration options. The newly created widget will instantly be accessible for all authenticated users to embed into their personal dashboards. Additionally, the widget creator may delete widgets.

3.3 Architecture and Implementation

To enable the embedding of the dashboard into arbitrary websites, and to allow the dashboard to connect to arbitrary data sources, the components of the dashboard infrastructure have been decoupled into a classic three-tier architecture—as depicted in Figure 2—consisting of a data layer, an application layer and a visualization layer. These layers are introduced in the following.

**Data Layer.** The data layer hosts a widget store database and a user preferences database for the dashboard, as well as an arbitrary number of relational databases provided by the embedding application, serving as the target community’s sources for the query result visualizations. Widget metadata stored on the data layer includes information on the available widgets which the end-users can add to their dashboard like title, description, categories, and location (URL).

**Application Layer.** The application layer offers components that allow the client-side components on the visualization layer to communicate with server-side components. While there are many potential existing application servers to chose from, we adopted the Lightweight Application Server (LAS) [4], which is a platform independent implementation of a lightweight middleware platform for service oriented architectures for the purpose of providing network services which can be shared among various tools supporting communities of practice. It offers functionality through three main elements: A LAS Connector provides the server-side interface for client-server communication using common communication protocols like HTTP or SOAP. Common server functionality like database access—e.g., for use in a LAS Service—is provided by LAS Components. For instance, the DB2 Component offers an interface to IBM DB2 databases for use in LAS Services. The actual functionality which the LAS server offers to its clients is encapsulated in LAS Services. The functionalities of a service are provided by its exposed service methods, which can be invoked by other services or clients. Two LAS services were developed to realize the dashboard functionality:

- The Dashboard Service, which offers methods to access the user and widget data on the data layer.

- The Query Service, which allows to query relational databases and return the query result in various output formats like XML, CSV or JSON. This functionality has originally been developed to visualize statistical data in order to analyze the success of mobile multimedia community services within the MobSOS project [3].
Visualization Layer. The visualization layer consists of two main components:

1. A dashboard container for widgets with extended controls to add and remove, move, search for and create widgets. The container was developed on the basis of common web technologies like HTML, CSS and JavaScript as an embeddable web application. Embeddable means that it requires only a few lines of HTML code plus some JavaScript files on the server-side to embed the dashboard into any web site. The dashboard currently supports the embedding of all widgets fulfilling the OpenSocial Gadget Specification [12]. Other widget specifications, for instance W3C Widgets [18], could be offered analogously. Through the LAS Ajax HTTP Client the client-side dashboard JavaScript code invokes the corresponding methods of the Dashboard Service on the application layer to retrieve, store and update visualization widget metadata and user preferences.

2. A visualization widget infrastructure to visualise query results as components of the dashboard. These widgets use the Query Service on the application layer to retrieve results for database queries. Using the Widget Creator Application (see Figure 4), any authorized user can create a new widget by providing the following information:

- Data source: the details of a database connection, by selecting from a set of pre-configured database connection strings including database host, port, user credentials and similar information.

- SQL query: will be sent to the selected database connection when the widget is embedded. In Figure 4 the user has already selected a database and is now previewing the pie chart visualization of her query.

- Visualization: defines how the query results will be visualized once the widget is embedded. The options currently include various chart types offered by the Google Visualization API [6], including table, bar chart, pie chart, line chart, radar chart, and annotated timeline. Additionally it offers a graph visualization of the query result. For radar charts and graphs the user may specify advanced visualization options.

When the user is satisfied with the visualization preview in the Widget Creator Application she can obtain a URL (button “Show Widget URL”) that encodes all information which the dashboard needs to display the visualization in the widget. Using this URL, the user may then complete a simple widget metadata form in the dashboard’s control pane to make this visualization available for all users to embed into their personal dashboard. In this step, the user provides the title, description and categories assigned to the widget. These pieces of information are displayed in the dashboard and support users in searching and browsing for interesting widgets.
Invocation Sequence. The invocations between components on the three layers are illustrated in Figure 2. A web application that embeds the dashboard has to supply two prerequisites, that is a page in which the dashboard is to be embedded, and the databases that contain the data sources for the visualization widgets. If the embedding application has its own user database, it must register a LAS user once for each of its users (invocation 1 in Figure 3) and hand over the current user’s credentials to the dashboard (2). All other invocations are sent and processed within the dashboard infrastructure:

- The dashboard logs in the user with his LAS credentials provided by the embedding application (3);
- The dashboard retrieves the list of available widgets from the widget database and displays these in the control pane (4);
- When a visualization widget is displayed in the dashboard (5a), it invokes the LAS Query Service (5b), which executes the specified query on the specified database connection (5c). The returned query result table is then transformed into a data table suitable for generating charts with the Google Visualization API (for examples see Figures 2 and 3);
- The dashboard stores—and restores—the current user’s dashboard preferences like the selection of embedded widgets, the column arrangement, and so forth (6).

4. PILOT APPLICATION

In this section we outline the application of the dashboard framework presented in the previous section in a community portal for Technology Enhanced Learning researchers and practitioners in Europe.

Background. In the PROLEARN project, an R&D project funded by the European Commission under FP6, one core effort was the creation and maintenance of a Mediabase for Technology Enhanced Learning (TEL), providing different stakeholders like scientists, policy makers, and communities of practice with digital information obtained from mailing lists, newsletters, blogs, RSS/Atom feeds, websites, and so forth [11]. These artifacts are continuously obtained from websites, blogs, wikis, and similar web sources. The data is updated by special components like importers and web crawlers which periodically retrieve relevant pieces of information from the observed sources and feed this information into the database(s). Besides the purpose of having a large amount of data available, one main objective of the Mediabase concept is the provision of easy to use end-user tools for extracting and presenting relevant information, e.g. for cross-media social network analysis, self-observation and self-modeling of communities [13], and collaborative administration and retrieval of media artifacts. The recently launched FP7 Support Action TEL-Map [http://telmap.org] adopted the Mediabase and extended the data sources with bibliographic data obtained from DBLP [http://dblp.uni-trier.de], and facts on collaborative EU funded projects obtained from the European Commission’s CORDIS portal [http://cordis.europa.eu].

Data Sources. In this application context the data layer includes the data storage (Mediabase) containing the data that can be queried and visualized in the dashboard. As available data sources three databases maintained by TEL-Map were connected to the dashboard: TEL Papers (Oracle), TEL Projects (MySQL) and TEL Social Media (DB2). These databases contain information on papers published in TEL-related journals and conferences, TEL related EU projects together with their participating organizations, and TEL related social media items like blogs or feeds, respectively [13]. Based on the information stored in the TEL-Map Mediabase, more than twenty queries addressing information demands of different TEL-Map stakeholders were formulated. For each of these queries, at least one visualization widget was created and added to the dashboard’s widget store. As an example consider the query visualization built in Figure 3, which offers TEL observers a pie chart visualization of EU funds spent in previous TEL related project calls.

Drupal Integration. The TEL-Map project is hosting the Learning Frontiers portal [http://learningfrontiers.eu], a roadmapping platform for the European Commission and TEL stakeholders. This portal is hosted on an instance of the Drupal content management system [http://drupal.org]. Drupal is extensible and customizable through Drupal modules. To offer the TEL stakeholders an aggregated, up-to-date view on data stored in the TEL-Map Mediabase, the dashboard was deployed as a Drupal module and connected to the TEL-Map Mediabase databases. Drupal maintains its own user database, while the dashboard uses LAS for user management. To shield users from having to login with multiple accounts, the Dashboard module in Drupal manages the automatic mapping of Drupal users to LAS users. Figure 4 shows a screenshot of the dashboard in the Learning Frontiers portal. The control pane is on the left, and the central widget container hosts two columns containing four widgets.

5. EVALUATION

Methodology. The dashboard implementation as integrated into the Learning Frontiers portal was evaluated with end-user involvement. A default dashboard with predefined visualization widgets based on the TEL-Map Mediabase was conceived for guest users. In addition, several widgets employing different visualizations of data contained in the three TEL-Map databases mentioned in the previous section were added to the widget database. The participants of the evaluation were consulted to visit the dashboard page, familiarize themselves with the dashboard, log in to the portal, and test each of the dashboard’s supported use cases by following a simple task sequence that was described in the evaluation form. After accomplishing a use case, the participants were asked to answer questions about their opinion. The evaluation questions addressed quantitative as well as qualitative factors. The former factors included the evaluation of the usability and the usefulness of the dashboard. The latter factors aimed at collecting open feedback and suggestions for improvement.

The questionnaire consisted of 20 questions; 18 of these questions were items with a standard five-point Likert scale ranging from strongly agree (5) to strongly disagree (1), and two questions asked for open text responses. Five additional questions were asked to collect some background information from participants, e.g. affiliation, years of experience...
The dashboard embedded in the Learning Frontiers portal with TEL, and so forth. The 18 quantitative questions were divided into two groups:

1. The ‘Usability’ group comprised nine questions targeting the evaluation of the general usability of the dashboard’s features;
2. The ‘Usefulness’ group surveyed participants’ opinion on the usefulness of dashboard and the visualizations in the dashboard’s widgets.

Participants. Participants were recruited mainly from TEL-Map project partners and registered users at the Learning Frontiers portal. In total the evaluation was completed by 20 participants from all over Europe. Of these, 17 were from universities, two from companies, and one from a TEL related organization. All but one participant work in a TEL related context, either as PhD students, TEL specialists or TEL researchers. The participants on average dedicate around two thirds of their total working time to TEL ($M = 65\%$, $SD = 33\%$) and rely on more than eight years of experience in TEL ($M = 8.55$, $SD = 7.94$).

Quantitative Survey Results. Overall, the participants have evaluated the dashboard positively ($M = 3.97$, $SD = .74$). Regarding the two question groups, the group ‘Usability’ ($M = 4.21$, $SD = .68$) achieved a noticeably higher average score than the group ‘Usefulness’ ($M = 3.72$, $SD = .84$). An overview of the survey results by question can be found in Figure 6. In this figure, the top nine bars with dark grey fill color represent the ‘Usability’ question group, while the bottom nine bars represent the ‘Usefulness’ question group.

To identify correlations between the years of experience with TEL and the rating of the dashboard a Pearson correlation analysis was performed. The result is that the more years of experience a participant can rely on, the worse he/she evaluated the dashboard overall ($r = -.623$, $p < .01$). Senior participants thus appear to have had higher expectations regarding the dashboard usability and the usefulness of visualized data, respectively.

Comparing all items, the question ‘This dashboard has all the functions and capabilities I expect it to have’ received the worst score ($M = 3.05$, $SD = 1.08$), and the question regarding the removal of widgets the best score ($M = 4.9$, $SD = 0.31$). This was not a surprising result, since removing widgets can be done with a single mouse click, while the user expectations regarding a TEL dashboard are expected to be multifaceted and thus difficult to satisfy, particularly with the participants in the sample having a strong TEL experience and high expectations towards the underlying data.

Qualitative Survey Results. To obtain qualitative feedback, the participants were asked two open questions. The first one referred to the user’s experience with creating a new widget. This use case was not expected to be performed by all participants, since it requires knowledge of the database schemas and SQL queries. Around half of the participants who answered this question considered the creation of widgets as easy (7). One participant failed during creating a widget, and three participants were able to successfully create a widget but the created widget produced an error message when the widget was embedded. Regarding improve-
### Survey Results

The survey results are presented in the following table and figures:

<table>
<thead>
<tr>
<th>Usability (M = 4.21 ± 0.68)</th>
<th>Usefulness (M = 3.72 ± 0.84)</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is easy to remove widgets</td>
<td>The data covered by available widgets appears relevant to me</td>
</tr>
<tr>
<td>It is easy to rearrange widgets</td>
<td>The data covered by the widgets appears relevant to the TEL community</td>
</tr>
<tr>
<td>It is easy to resize the widget height and width</td>
<td>The available widgets cover a broad range of TEL</td>
</tr>
<tr>
<td>It is easy to detach widgets</td>
<td>The available categories are useful to search for widgets</td>
</tr>
<tr>
<td>It is easy to find the description of a widget</td>
<td>The style of the data visualizations is appropriate</td>
</tr>
<tr>
<td>The widget descriptions are sufficient to understand the widget content</td>
<td>The range of available data visualizations is sufficient</td>
</tr>
<tr>
<td>The search interface is easy to use</td>
<td>This dashboard has all the functions and capabilities I expect it to have</td>
</tr>
<tr>
<td>It was easy to find relevant widgets</td>
<td>The dashboard is useful to support mapping and roadmapping activities in TEL</td>
</tr>
<tr>
<td>It was easy to add available widgets</td>
<td>The dashboard can support different TEL stakeholders in observing data</td>
</tr>
</tbody>
</table>

**Figure 6: Survey results**

Table entries are ranked from 1 to 5, where 5 indicates the highest level of agreement. The table includes statements such as clean and intuitive visualizations, useful tool, difficulties with showing the control pane, missing access to data behind a visualization, just useful for statistics about TEL, problems with understanding the meaning of some categories, difficulties with understanding the meaning of some categories, detaching does not give more space for the visualization, and complex visualizations are hard to read.

### Action Log Analysis

During the evaluation period (13 days in total) data traces for each action performed by a user with the dashboard were collected and stored in an action log table in the widget database. This included actions like searching for widgets, rearranging a widget, adding a widget to the dashboard, and so forth. The action logging not only included the 20 users who managed to submit an evaluation form, but also five users who ‘dropped out’ during evaluation.

In total, 687 actions were logged, that is an average of 27 actions per user. The most frequently performed action was toggling the visibility of the control pane (196), followed by adding an existing widget to the dashboard (141). The removal of a widget from the dashboard was invoked 101 times. The drag-and-drop customization functionalities were collectively used 142 times: move widget (88), resize widget (28), and resize column (26). While browsing for widgets, the participants made use of the category filter 50 times and of the keyword search for 30 times. The number of columns was adjusted only 10 times. In total, the participants created 17 new widgets during the evaluation, most of which were existing OpenSocial compliant widgets, and not visualizations of database queries.

Of the 141 widget embedding actions, 20 belonged to the category ‘Aggregate’, followed by the categories ‘Comparison’ (18) and ‘Categories’ ‘Papers’ and ‘People’ (15). The most frequently embedded chart type is the timeline chart (17), followed by the network graph (15). The bar chart was embedded 14 times, the pie chart 12 times and the radar chart 11 times. The most frequently embedded widget visualizes the TEL blogosphere activity over the last three months as a timeline chart; ten users embedded this widget.

### Summary

In general, the evaluation shows that the dashboard generally managed to meet the expectations of the consulted TEL experts with regard to usability and usefulness. Usability scores were generally higher than useful-
ness scores and both correlate negatively with the number of years of TEL expertise of the participant. The responses to the open questions indicate that there is potential for improvements to the dashboard, both regarding the provided functionality and the visualization of the underlying data. However, the usefulness of the underlying data is somewhat out of the scope of this paper, since these data sources are maintained by the TEL-Map project independently of the dashboard infrastructure presented here.

6. CONCLUSION

In this paper we described the concept, development and evaluation of a user friendly and effective infrastructure for visually interacting with large amounts of data through a widget-based dashboard that can be embedded in arbitrary web sites. To support visual analytics tasks the dashboard provides a personalizable interface for monitoring of the data, stored in different databases based on the visualization of SQL query results in widgets using the Google Visualization API. The personalization and customization options enable the users to select particular visualization widgets to embed in their dashboard as well as the modification of the appearance of the dashboard view. One main objective of the dashboard architecture was to make the dashboard embeddable in arbitrary web sites and connect to arbitrary databases using open web technologies that are implemented in all popular web browsers.

As a demonstration of (1) the technical feasibility of the proposed dashboard infrastructure and (2) the usability and usefulness of the dashboard functionality to end-users, the dashboard was integrated into the Learning Frontiers portal, a Drupal based website which is hosted by the TEL-Map project. The dashboard was connected to the TEL-Map Mediabase, a collection of three relational databases containing data on social media, publications and projects in the Technology Enhanced Learning (TEL) domain.

Technically, the integration of the dashboard in Drupal was straightforward and did not present any problems. From a user perspective, the dashboard embedded in the Learning Frontiers portal was evaluated by Europe wide involvement of TEL experts and stakeholders. Although the dashboard was generally evaluated positively based on the quantitative items regarding usability and usefulness in the evaluation survey, the qualitative free-text items showed that the dashboard offers space for improvements, for instance regarding data export and visual arrangement of widgets. Also, further work is needed regarding the actual support of visual analytics tasks. Visualizations are only one aspect of facilitating analytics, and the predefined queries are rather inflexible in supporting explorative tasks, reasoning and decision making, as evident in the participants’ feedback. The visualization features and the guidance in using the dashboard need to be more tightly intertwined with analytical tasks carried out by scientific community members in their daily work.

Regarding the usefulness of the dashboard, the integration into the Learning Frontiers portal has indicated that the users’ perceived benefits of such a dashboard strongly rely on the quality of the data it provides, which came as no surprise. To support the users in their daily visual analytics work, the dashboard would certainly have to offer a larger and more diverse set of high quality visualizations. Nonetheless, the study provided further evidence that widget-based dashboards are promising visual analytics tools for monitoring and awareness within scientific communities, particularly when these tools enable a lightweight integration with the communities’ data sources and web portals.

7. ACKNOWLEDGMENTS

This research was funded by the European Commission through the TEL-Map Support Action (FP7-257822). We thank Dominik Renzel, Lucas Dohmen, Malte Behrendt, and Michael Hackstein for their help during implementation.

8. REFERENCES