Designing Generic Visualisations for Activity Log Data

Granit Luzhnica[1], Angela Fessl[1], Eduardo Veas[1], Belgin Mutlu[1], Viktoria Pammer[2]

[1] Know-Center, Inffeldgasse 13
A - 8010 Graz
(gluzhnica, afessl, eveas, bmutlu)@know-center.at
[2] Graz Univ. of Technology, Inst. of Knowledge Technologies, Inffeldgasse 13
A-8010 Graz
viktoria.pammer@tugraz.at

Abstract. Especially in lifelong or professional learning, the picture of a continuous learning analytics process emerges. In this process, heterogeneous and changing data source applications provide data relevant to learning, at the same time as questions of learners to data change. This reality challenges designers of analytics tools, as it requires analytics tools to deal with data and analytics tasks that are unknown at application design time. In this paper, we describe a generic visualization tool that addresses these challenges by enabling the visualization of any activity log data. Furthermore, we evaluate how well participants can answer questions about underlying data given such generic versus custom visualizations. Study participants performed better in 5 out of 10 tasks with the generic visualization tool, worse in 1 out of 10 tasks, and without significant difference when compared to the visualizations within the data-source applications in the remaining 4 of 10 tasks. The experiment clearly showcases that overall, generic, standalone visualization tools have the potential to support analytical tasks sufficiently well.

1 Introduction

Reflective learning is invaluable for individuals, teams and institutions to successfully adapt to the ever changing requirements on them and to continuously improve. When reflective learning is data-driven, it comprises two stages: data acquisition and learning analytics. Often, relevant data is data about learner activities, and equally often, relevant activities leave traces not in a single but in multiple information systems. For instance, [14] presents an example where activities relevant for learning about software development might be carried out in svn, wiki and an issue tracking tool. In the context of lifelong user modeling, the learning goals and learning environments change throughout life, different software will be used for learning, while the lifelong (open) user model needs to store and allow analysis across all collected data [12]. Furthermore, as ubiquitous sensing technologies (motion and gestures, eye-tracking, pulse, skin conductivity, etc.) mature and hence are increasingly used in learning settings, the data sources for such standalone learning analytics tools will include not only information systems but also ubiquitous sensors (see e.g., [24] or [2] who calls this “multimodal learning analytics”). Furthermore, it is frequently the case that concrete data sources, and consequently the questions that users will need to ask of data (analytic tasks) are not a priori known at the time of designing the learning analytics tools. In the context of lifelong learning for instance, at the time of designing a visualization tool, it cannot be foreseen what kind of software will be used in the future by the learner. In the context of the current trend towards rather smaller learning systems (apps instead of learning management systems) it is plausible to assume that learners may wish to exchange the used software regularly (and be it only that they switch from Evernote to another note-taking tool). At the extreme end of generic analytics tools are of course expert tools like SPSS and R, or IBM’s ManyEyes [25] for visualizations.

A picture of a continuous learning analytics process emerges, in which heterogeneous and ever changing data source applications provide relevant data for learning analytics, at the same time as questions of learners to data also continuously change. To support such a continuous analytics process, we have developed a generic visualization tool for multi-user, multi-application activity log data. In this paper, we describe the tool as well as the results of a task-based comparative evaluation for the use case of reflective workplace learning. The generic visualization tool integrates data from heterogeneous sources in comprehensible visualizations. It includes a set of visualizations which are not designed for specific data source applications, thus the term *generic*. It can visualize *any* activity log data published on its cloud storage. The only prior assumptions are that every entry in the data should be: i) time stamped and ii) associated with a user. The tool thus strikes a balance between generality (few prior assumptions) and specificity.

One key concern was whether the developed generic visualizations tools would be as comprehensible as those designed specifically for a given application or dataset. In this paper we describe an experiment comparing the performance of study participants along learning analytics tasks given the generic visualizations and visualizations custom-designed for the respective data.

2 Related Work

Others before us have pointed out the need to collect and analyze data for learning across users (multi-user) and applications (multi-application), both in the community of learning analytics and open learner modeling: Learning analytics measures relevant characteristics about learning activities and progress with the goal to improve both the learning process and its outcome [16,23]. Open learner models collect and make intelligible to learners and in some use cases also to peers and teachers data about learning activities and progress as well, again as basis for reflection on and improvement of learning [4]. Also in user modeling, the visualization of data across users is a relevant topic (e.g., [14]). Clearly, relevant characteristics about learning activities reside very rarely only in a single system, and both communities have identified a need to collect and analyze data from heterogeneous data sources [12,18]. For instance, in Kay and Kummerfield [13] a variety of external data sources (mainly health sensor's data) is used for aggregation, analysis and visualization (through external applications) to support completing Sisphean tasks and achieving long term goals.

Visualizations in learning analytics and open learner modeling play the role of enabling users (most often students, teachers, but also institutions or policy makers - cf. [23]) to make sense of given data [6]. Most papers, however, predefine the visualizations at design time, in full knowledge of the data sources [6,10,14,15,20,21]. In Kay et al. [14] for instance, teams of students are supported with open (team) learner models in learning to develop software in teams. The authors developed a set of novel visualizations for team activity data, and showed the visualizations' impact on team performance (learning). In Santos et al. [22], student data from Twitter, blog posts and PC activity logs are visualized in a dashboard. The study shows that such dashboards have a higher impact on increasing awareness and reflection of students who work in teams than of students who work alone. Again, data sources are defined prior to developing visualizations however. In such visualizations, users "simply" need to understand the given visualizations, but do not need to create visualizations themselves.
On the other end of the spectrum are extremely generic data analytics tools such as spreadsheets or statistical analysis tools like SPSS or R. Outstanding amongst such tools is probably IBM's web-based tool ManyEyes. Users can upload any data at all in CSV format, label and visualize data. ManyEyes makes no assumptions at all about uploaded data, but clearly puts the burden of figuring out what kind of visualizations are meaningful to the users.

3 Generic Visualization Tool for Activity Log Data

We have developed a generic visualization tool for activity log data that addresses two fundamental challenges shared in many scenarios at the intersection of learning analytics, open learner modeling, and reflective learning on the basis of (activity log) data: Data from multiple applications shall be visualized; and at the time of designing the visualization tool, the concrete data sources and consequently the concrete analytic tasks are unknown.

3.1 A Priori Knowledge about Data

We make only two assumptions about data, namely that they are i) time stamped and ii) every data entry is associated with a user. The second assumption is useful because in a lot of learning scenarios, learning is social [1]: Students regularly work in teams as well as employees in organizations (in the context of workplace learning). Depending on the applications that are used to collect the activity log data, and the users' sharing settings, data from other users may be available. Therefore, it is reasonable to assume that meaningful insights can be gained by analyzing not only data from one individual but also data from multiple users.

3.2 System Architecture and Data Format

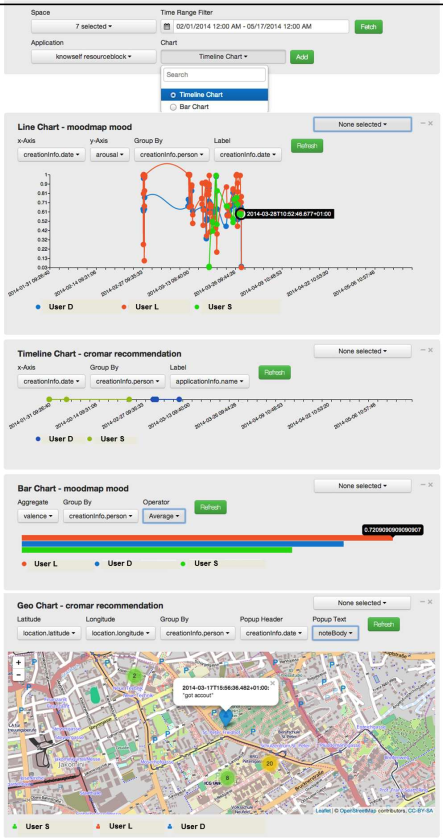
The generic visualization tool (Vis-Tool) is implemented as a client-server architecture. It is a web application implemented in HTML5/Javascript and Java. The Vis-Tool itself does not capture any activity log data, but is responsible for the visualization of the data in a sophisticated and meaningful way. Through its server component, it is connected to a could storage that stores application data and manages access to data. The data source applications store their activity log data on the cloud storage in so-called spaces: Private spaces store data of only one user, while shared spaces collect data from multiple users. Single sign-on provides a common authentication mechanism for all data-source applications, the Vis-Tool and the cloud storage. The rationale behind this chosen architecture is to deal with data collection, data analysis and data visualization separately.
The Vis-Tool expects data in an XML format described by a publicly available XML schema. In addition, the schema must extend a base schema that contains a unique ID for all objects, a timestamp and a user ID as mandatory fields.

3.3 Single-Column Stacked Visualization

The Vis-Tool organizes visualizations in form of a dashboard style similar to [6,21,15,10], but we use a single column for the visualizations. Visualizations are always stacked on top of each other and share the same time scale, whenever possible. This is necessary to directly compare the data from different applications along the very same timeline (see Fig. 1). Users can add charts to their dashboard using an ”Add” button. Charts can be minimized (”-” button) or completely removed (”x” button), which are located at the top right corner of each chart. The position of each chart can be rearranged by using drag and drop. Thus, a user can easily adapt the visualizations to one's individual needs.

3.4 Chart Types

The Vis-Tool provides four types of charts with different visual channels: geo chart, bar chart, line chart and timeline chart (see Figure 1).
The geo chart is used for data that contains geo positions. Besides the ”latitude” and ”longitude”, the chart consists also of the “popup header” and “popup text” as additional visual channels. Both of them are shown in a popup window when clicking on an entry. The bar chart is available for any structure of data. It contains the “aggregate” channel and the “operator” setting. While the “aggregate” channel defines, which data property should be aggregated, the “operator” defines how the data will be aggregated (count, sum, average, min max) in order to be displayed. The line chart contains “x-axis”, “y-axis”, and “label” (on hover text). It is available for data with numerical data properties. Our timeline chart is similar to the line chart but does not have an “y-axis” channel. All charts have the “group by” channel. It defines how data can be grouped with the help of colors. For example, if we use a user id to group the data belonging to one user, all data captured by this user will be presented with the same color. If



Space: 7 selected
Time Range Filter: 02/01/2014 12:00 AM - 05/17/2014 12:00 AM
Application: knowself resourceblock
Chart: Timeline Chart

Line Chart - moodmap mood
x-Axis: creationInfo.date
y-Axis: arousal
Group By: creationInfo.person
Label: creationInfo.date

Timeline Chart - cromar recommendation
x-Axis: creationInfo.date
Group By: creationInfo.person
Label: applicationInfo.name

Bar Chart - moodmap mood
Aggregate: valence
Group By: creationInfo.person
Operator: Average

Geo Chart - cromar recommendation
Latitude: location.latitude
Longitude: location.longitude
Group By: creationInfo.person
Popup Header: creationInfo.date
Popup Text: noteBody

2014-03-17T15:56:36.482+01:00:
"got accout"

Fig. 1: Vis-Tool user interface (at the top) including four charts with user data in the single column dashboard.

several users are added to a group, all data captured by the users belonging to this group, will be presented with the same color. This feature makes it possible to easily spot user patterns across applications.

3.5 Mapping Data to Visualizations

Users create charts in the Vis-Tool by selecting data they wish to visualize, selecting a chart type, and then filling the chart's visual channels. The Vis-Tool, however, presents only those options to the user that are possible for any given data. Technically this is solved with chart matching and channel mapping.

3.5.1 Chart Matching For each chart type, we created a chart description consisting of a list of all channels, mandatory as well as optional, and the data types that the channels can visualize. At runtime, the XML schemas that describe the structure of the user data are parsed and the data properties including their data types are extracted. Based on the extracted data structures and the available channels per chart, chart matching is performed. The matching determines whether a chart is able to visualize a dataset described by a parsed schema.

This is done by checking for each channel of the chart, if the data structure has at least one property whose data type can be visualized by the given channel. For instance, the line chart consists of x-axis, y-axis as mandatory channels and the hover text as optional channel. The x-axis can visualize numeric values and date-time values and the y-axis can handle numeric values. The hover text channel is able to handle numeric values, date-times and strings. The line chart will be available for the parsed data structure, if the structure contains at least one numeric type or date-time for the x-axis and a numeric type for the y-axis. The hover text is an optional channel and therefore not of relevance for the decision, if the line chart is able to present the parsed data structure or not. For a given data structure, chart matching is performed for each chart type. Those chart types that match with the given data structures are added to the list of possible chart types and can be selected.

3.5.2 Channel Mapping Channel mapping takes place if a user selects one of the available charts. An initial channel mapping is automatically provided to the user when adding a chart to the visualization. Users can adapt the mapping of a property to another chart channel via the interface.

4 Use Case

The Vis-Tool can be used in any use case in which analysis of multi-user and multi-application activity log data makes sense. A lot of learning analytics and open learner modeling use cases fall into this category, as argued above. The task-based comparative evaluation that we subsequently describe and discuss in this paper assumes a specific use case however. It is one of knowledge workers who work in a team, carry out a significant amount of their work on desktop PCs, and spend a significant amount of time traveling. In the sense of reflective work-integrated learning [3,7] knowledge workers would log a variety of aspects of their daily work, and routinely view the log data in order to gain insights on their working patterns and change (for the better) their future working patterns. Concretely, we evaluate the Vis-Tool in comparison to three specific activity log applications that all have been successfully used and evaluated in the context of such reflective workplace learning [5,8,17].

Collaborative Mood Tracking - MoodMap App[3] [8] - is a collaborative self-tracking app for mood, based on Russell's Circumplex Model of Affect [19]. Each mood point is composed of "valence" (feeling good - feeling bad) and "arousal" (high energy - low energy). The mood is stated by clicking on a bi-dimensional mood map colored according to Itten's system [11]. Context information and a note can be manually added to the mood, while the timestamp is automatically stored. Depending on the user's setting, the inserted mood is kept private or shared with team members. Mood is visualized on an individual as well as collaborative level. The MoodMap App has been successfully used in virtual team meetings to enhance team communication by inducing reflection [8].

[3] http://know-center.at/moodmap/

Example analysis on MoodMap data for workplace learning are to review and reflect on the development of individual mood in comparison to team mood, and in relationship to other events or activities that happen simultaneously.
PC Activity Logging - KnowSelf[4] [17] automatically logs PC activity in the form of switches between windows (associated with resources like files and websites as well as applications). Manual project and task recording, as well as manually inserted notes and comments complete the data captured by the app. The visualizations are designed to support time management and showcase in particular the frequency of switching between resources, the time spent in numerous applications, and the time spent on different activities. KnowSelf has concretely been used as support for improving time management [17], but activity logging data has also been used as basis for learning software development in an educational context [21,22]. Example analyses of PC activity log data for workplace learning are to relate time spent in different applications to job description (e.g., the role of developer vs. the role of team leader), and to relate the time spent on recorded projects to project plans.
Geo Tagged Notes - CroMAR [5] is a mobile augmented reality application designed to show data that was tagged with positions around the user's place. The information is overlayed on the video feed of the device's camera. CroMAR allows users to create geo-tagged data such as notes and pictures. The notes are stored in the cloud storage. CroMAR has features that are relevant for reflecting on any working experience with a strong physical nature. It was specifically developed for reflection on emergency work, in particular in relation to crowd management. CroMAR has been evaluated in the domain of civil protection to review, location-based, what happened during events [5]. A typical use case for knowledge workers would be to reflect both on the content of notes, and their relation to particular locations (which would typically be in line with customers, project partners, location-based events, or travel-related locations).

4.1 The Potential Benefit of Combining Data Across Applications

In prior work, we explored the potential benefit of analyzing data from PC activity logging data together with collaborative mood tracking data in such a use case [9]. As one example, a user's mood might drop consistently in relation to a particular project. In addition, we conjecture that mood might also be related to particular places, or some kinds of work might be carried out more productively outside the office.

5 Research Approach: Task-Based Evaluation

We performed an evaluation to compare custom visualizations in the data source applications (in-app) with generic visualizations (Vis-Tool). The goal was to establish how the comprehensibility of generic visualizations, designed without specific prior knowledge about (meaning of) data, compares to custom in-app visualizations that were customized for a specific kind of data and task.

[4] http://know-center.at/knowself/

5.1 Data preparation

We prepared a test dataset with data about three users, called **D**, **L** and **S**, containing two weeks of data from all applications. To do so, we extracted data from real usage scenarios of the single applications. For MoodMap, we selected two weeks of the three most active users out of a four-week dataset. For KnowSelf, we selected three two-week periods of log data out of a 7-month dataset from a single user. For CroMAR, we used the dataset from a single user who had travelled significantly in a two-seeks period, and manually created two similar datasets to simulate three users. The data were shifted in time so that all datasets for all applications and users had the same start time and end time.

5.2 Evaluation Procedure

The evaluation is intended to test the comprehensibility of generic visualizations for learning analytics. We wanted to investigate how understandable are generic visualizations compared to the custom visualizations that are specifically designed for data of one specific application. Our initial hypothesis was that the generic visualizations could be as meaningful as custom visualizations. As we wanted to rule out confounding factors from different interaction schemes, we opted to perform the experiment on paper based mock-ups. These were created from the datasets by capturing screenshots of the in-app visualizations and the generic ones generated with the Vis-Tool. We prepared short analytics tasks (see Table 1) that participants should solve with the use of the given visualizations. The tasks are plausible in line with the chosen use cases (see Section 4) above, which were constructed based on use cases of knowledge workers that were previously evaluated in their working environment [8,17] as well as use case exploration of joint data analysis [9]. We simulated the hover effect, clicking, scrolling and zooming by first letting the participants state the action and then replacing the current mockup with a new corresponding one.

The evaluation followed a within-participants design. For each tool (MoodMap App (MM), KnowSelf (KS), CroMAR (CM)) we created a number of tasks (MM=4, KS=4, CM=2). We created variants of each task with different datasets for each condition (Vis-Tool, in-app). Thus, there were 20 trials per participant (10 tasks in 2 conditions for each participant). Tasks and tools were randomized across participants to avoid favoring either. We grouped the tasks by tool and randomized the order of groups, the tasks within groups and the order of condition (in-app visualization / generic visualization). The experimenter measured the duration (time to completion) and real performance for each task. Additionally, subjective scores of difficulty were measured through self-assessment using the NASA-TLX workload measure [9]. The tasks were organized in groups, each containing tasks with data generated from a single log activity application. Table 1 summarizes the tasks per tool.

The study followed the format of a structured interview, where the experimenter first explained the goals, the applications and the tasks participants would perform. The participant then proceeded to the first task, which finalized

T#	App	Task
1	MM	On the given day, to whom did belong the worst single energy (arousal) and to whom did belong the single worst feeling(valence)?
2	MM	On the given day, to whom did belong the worst average energy (arousal) and to whom did belong the worst average feeling (valence)?
3	MM	Find out on which day in the two recorded weeks was entered the best energy(arousal) and best feeling (valence) of the user!
4	MM	Find out on which days (dates) the MoodMap App was not used at all!
5	KS	On given day, when exactly (at what time) did the given user had the longest break? How long was the break?
6	KS	Find out on which day in the two recorded weeks did **L** work (regardless of breaks) longest?
7	KS	Find out which application was most frequently used in the last two weeks by given user!
8	KS	Find out which user used MS Word most often on the given day!
9	CM	**(a)** Find out where (in which Countries) in Europe have notes been taken! **(b)** Find out in which cities in Austria did **L** and **D** take notes!
10	CM	**(a)** Find out how many notes have been created at Inffeldgasse, Graz! **(b)** Find out how many notes have been created in Graz!

Table 1: Tools and evaluation tasks. **L**, **D** and **S** are the initials of the users to whom the data belong.

with the NASA-TLX. After finishing each group a questionnaire was distributed to directly evaluate the visual design, comprehensibility and user preference of in-app visualizations in comparison to the Vis-Tool visualizations.

5.3 Participants

Eight people participated in the experiment, all knowledge workers (researchers and software developers). Three of them were female, 5 male. 3 participants were aged between 18-27 and 5 were aged between 28-37.

6 Results

Overall, our study participants performed significantly better with the generic visualization tool in five (T2, T3, T7, T9, T10) out of ten tasks, worse in only one (T5) task and without significant difference when compared to the in-app visualizations in the remaining four (T1, T4, T6, T8) tasks. To analyze results, the Fisher's Test was used to check the homogeneity of variances. The tf-test was used to test significance for cases with homogeneous variance. If not, the Walch-Satterthwaite test was used.

6.1 Workload

The NASA-TLX includes six metrics, which are considered scales of workload. We used the simplified R-TLX method to compute workload by averaging the scores. Figure 3 (MoodMap vs. Vis-Tool), Figure 4 (KnowSelf vs. Vis-Tool)

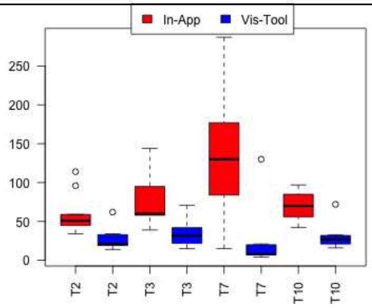


Fig. 2: Task duration (in seconds) for all tasks with significant differences.

and Figure 5 (CorMAR vs. Vis-Tool) show the box plots of the significant results for NASA-TLX metric: mental demand (MD), physical demand (PD), temporal demand (TD), measured performance (MP) and frustration (F) as well as the workload (W), computed as the average of all self-evaluation metrics and the measured performance (MP). Task duration (D) for all apps is given in Figure 2. The result of the t-test for T2 indicates that participants

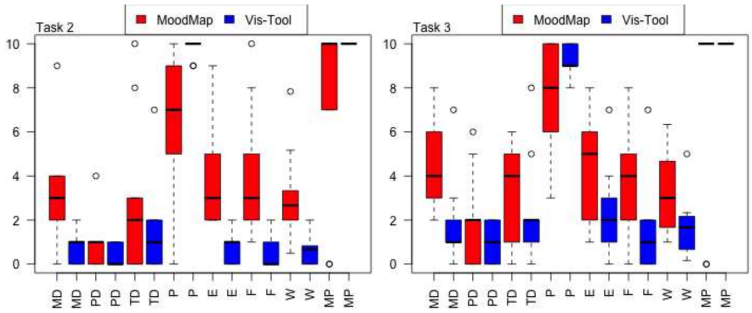


Fig. 3: MoodMap vs. Vis-Tool (T2,T3) - Significant NASA-TLX results.

experienced significantly less workload when using Vis-Tool than MoodMap, $t(9) = 3.17; p < .01$. Also, the task duration was significantly lower in the case of Vis-Tool, $t(9) = 3.18; p < .01$. In fact all individual metrics show significantly better scores in favor of Vis-Tool. For T3, there was a significant less workload and significant less duration when using Vis-Tool, $t(9) = 2.13; p < .05$ respectively $t(9) = 3.44; p < .01$. For T5, there was a significantly lower workload when

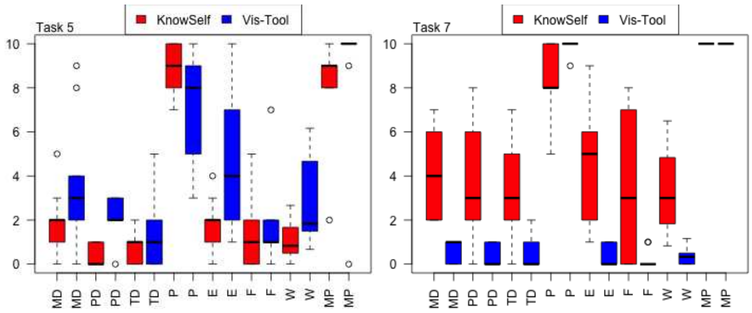


Fig. 4: KnowSelf vs. Vis-Tool (T5,T7) - Significant NASA-TLX results.

using KnowSelf in comparison to Vis-Tool, $t(9) = 2.21; p < .05$. Individual metrics show a significant difference in effort and physical demand (see Figure 4). For T7, except for measured performance (MP), significant differences were found in every other metric. Participants experienced significantly lower workload using Vis-Tool, $t(9) = 4.60; p < .01$. They also spent significantly less time solving the task with Vis-Tool, $t(9) = 3.64; p < .01$. In the group CroMAR VS Vis-Tool, the results of both tasks show significant differences in favor of the Vis-Tool (see Figure 5). For T9, there was a significant difference in measured performance, $t(9) = 3.16; p < .02$. Individual metrics show significant difference in mental demand. For T10, there was a significantly less workload when using Vis-Tool, $t(9) = 2.36; p < .04$. Analysis of individual metrics showed significant differences in mental and physical demand. Duration was also significantly different in favor of Vis-Tool, $t(9) = 4.68; p < .01$.

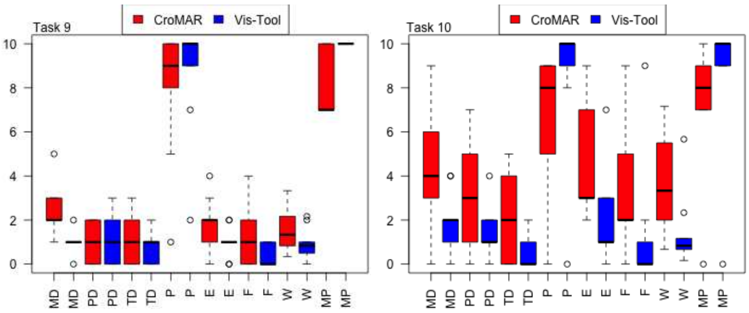


Fig. 5: CroMAR vs. Vis-Tool (T9,T10) - Significant NASA-TLX results.

6.2 Application Preferences and Visual Design

The summarized results of the user preferences regarding the used apps for solving the given tasks are presented in Table 2. For the tasks T1-T4 and T9-T10 Vis-Tool was preferred over both MoodMap and CroMAR. For the tasks T5-T8 the results of Vis-Tool vs. KnowSelf were ambiguous. For T5 and T6 participants preferred KnowSelf whereas for the tasks T7 and T8 they go for the Vis-Tool. This correlates with TLX where users performed better using KnowSelf in T5 but much worse in T7.

The results of the question “How did you like the visual design of the visualisations for the given tasks?” (see Figure 6) showed a clear preference for the visual design of the Vis-Tool in comparison to the MoodMap (tasks T1-T4) and CorMAR (tasks T9-T10). In contrast, for the tasks T5-T8 they preferred the visual design of KnowSelf over that of the Vis-Tool. Regarding the question “How meaningful were the given visualizations for the given tasks?” the participants stated that Vis-Tool visualizations where significantly more meaningful for the given tasks in comparison to the MoodMap and CroMAR (see Figure 6). Interestingly, there were no significant results regarding Vis-Tool and KnowSelf.

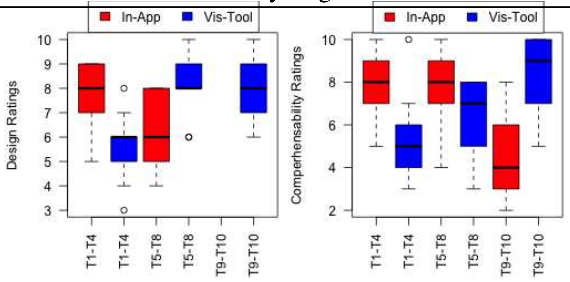


Fig. 6: User ratings on the design and comprehensibility of the visualizations.

	T1	**T2**	**T3**	**T4**	**AVG**	**T5**	**T6**	**T7**	**T8**	**AVG**	**T9**	**T10**	**AVG**
Vis-Tool	89%	78%	67%	44%	69%	22%	22%	100%	78%	56%	67%	100%	83%
In-app	11%	0%	0%	0%	3%	67%	56%	0%	11%	33%	33%	0%	17%
Both	0%	22%	11%	11%	11%	11%	11%	9%	9%	6%	0%	0%	0%
None	0%	0%	22%	44%	17%	0%	11%	0%	11%	6%	0%	0%	0%

Table 2: Which visualizations are preferred for solving the given tasks?

7 Discussion

Overall, the performance of study participants was satisfactory with the Vis-Tool, showing comparable and mostly even better performance when compared with in-app visualizations. In many cases, study participants had a significantly lower workload and were significantly quicker to solve the tasks using generic visualizations: Participants achieved significantly better results with the Vis-Tool than with the MoodMap App in two out of four tasks in terms of workload and time to task completion (T2, T3 - see also Figure 2 and 3), better results with the Vis-Tool than with KnowSelf in one out of four tasks (T7 - see also Figure 2 and 4) and better results with the Vis-Tool than with CroMAR in two out of two tasks in terms of task performance (T9) and workload and duration (T10 - see also Figure 2 and 5). These results are also confirmed by the answers of the questions regarding the comprehensibility of the visualizations with regard to the given tasks (see Table 6).

7.1 Supporting Analytic Tasks Beyond Design Time

These results are not a statement on the quality of design of the specific apps per se. All three used activity logging applications have successfully been used to induce and support learning in the workplace. Rather, the results are a function of whether the data source applications have been designed to answer the type of questions about data that study participants were asked to answer in the evaluation. The focus of CroMAR for instance was in location-related, augmented-reality-style, visualization of geo-tagged data in order to support situated reflection on events [5]. Quite naturally then, its user interface is less conducive to answering general questions about data. The focus of KnowSelf on the other hand was to support users in reviewing their time use daily in order to support time management [17]. This is visible in the comparative results which show a strong task dependence: Participants find it easier to perform the task that relates to a single day (T5) with KnowSelf than with the Vis-Tool, but find the

Vis-Tool more supportive in a task that relates to a longer period of time (T7). Another example of generic visualizations adding benefit to in-app visualizations is that the data source applications had different support for multiple users: KnowSelf is a purely single-user application; nonetheless, there is a plausible interest within teams to know how others in the team use their time. CroMAR visualizes data from multiple users but does not visually mark which data comes from which user, and MoodMap App is a real collaborative tracking application. Our study results therefore clearly showcase that and how generic visualizations can add benefit to in-app visualizations when users want to solve analytic tasks beyond those that were known at application design time.

7.2 Visualizing Derived Data Properties

A limitation of the current implementation of the Vis-Tool is, that it is only able to display given properties, but cannot calculate new values. For instance, in KnowSelf, the data entries contain the start and the end time but not the duration. The visualizations in KnowSelf make use of such derived data properties: As KnowSelf developers know exactly what kind of data were available, they could also easily implement calculations based on given data and use these for visualizations. In the Vis-Tool on the other hand, we have in general too little prior knowledge about data to automatically perform meaningful calculations on data in order to compute "derived data properties". Technically, it would be possible to extend the Vis-Tool's user interface such that calculations on given data can be specified, but we assume that ease of use would be rather difficult to achieve. In addition, such functionality would increasingly replicate very generic spreadsheet (e.g., Excel), statistical analysis (e.g., SPSS) or visualization (e.g., ManyEyes) functionality. It might be easier overall to shift the burden "back" to data source applications, in the sense of requiring data source applications to provide derived values that are of interest themselves.

7.3 Ease of Interaction

In this work we have focused on the comprehensibility of visualizations. We did not formally evaluate the user interaction itself, i.e. the process of creating a specific visualization. However, we are aware that the Vis-Tool requires users to become familiar with concepts such as mappings and visual channels.

A plausible emerging scenario is to differentiate between two user roles: One role (expert) would be responsible for creating a set of meaningful visualizations. The expert would know concretely which data source applications are available and what kind of analytic tasks users will want to solve. This person does not need to write code, but needs to have some training or experience with the Vis-Tool. The set of meaningful visualizations would be stored and serve as pre-configuration for learners. A second role (learner) would then only need to load a pre-configured set of visualizations and "use" them, similar to the study participants in the task-based evaluation discussed in this paper. Of course, users would have the freedom to explore the mapping interface if interested, and generate new visualizations. Based on this overall scenario, more complex usage scenarios for generic visualization tools like ours could be elaborated that involve for instance sharing and recommending dashboards.

8 Conclusion

We have developed a generic visualisation tool for activity log data that addresses two fundamental challenges shared in many scenarios at the intersection of learning analytics, open learner modelling, and workplace learning on the basis of (activity log) data: Data from multiple applications shall be visualised; and at the time of designing the visualisation tool, the concrete data sources and consequently the concrete analytic tasks are unknown. The Vis-Tool makes only two assumptions about data, namely that they are time-stamped and are associated with users. The comprehensibility of the Vis-Tools visualisations was evaluated in an experiment along data analytics tasks that were designed on the background of workplace learning. This evaluation was carried out within the target user group of knowledge worker, and based on real-world data. It thus constitutes firm ground, also for other researchers, to compare the suitability of other generic visualisations with, or to proceed with the next step in the design process for such a generic visualisation tool, namely the design of the user interaction process.

Acknowledgments

The project "MIRROR - Reflective learning at work" is funded under the FP7 of the European Commission (project nr. 257617). The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. A. Bandura. *Social Learning Theory.* General Learning Press, New York, 1977.
2. P. Blikstein. Multimodal learning analytics. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 2013.
3. D. Boud, R. Keogh, and D. Walker. *Reflection: Turning Experience into Learning*, pages 18–40. Routledge Falmer, New York, 1985.
4. S. Bull and J. Kay. Student models that invite the learner: The smili:open learner modelling framework. *International Journal of Artif. Intell. in Education*, 2007.
5. M. Divitini, S. Mora, and A. Boron. Cromar: Mobile augmented reality for supporting reflection on crowd management. *Int. J. Mob. Hum. Comput. Interact.*, 4(2):88–101, Apr. 2012.
6. E. Duval. Attention please!: learning analytics for visualization and recommendation. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 2011.

7. M. Eraut and W. Hirsh. The Significance of Workplace Learning for Individuals, Groups and Organisations, SKOPE Monograph 9, Oxford University Department of Economics, 2007.
8. A. Fessl, V. Rivera-Pelayo, V. Pammer, and S. Braun. Mood tracking in virtual meetings. In *Proceedings of the 7th European conference on Technology Enhanced Learning*, EC-TEL'12, pages 377–382, 2012.
9. A. Fessl, G. Wesiak, and G. Luzhnica. Application overlapping user profiles to foster reflective learning at work. In *Proceedings of the 4th Workshop on Awareness and Reflection in Technology-Enhanced Learning (Colocated with ECTEL)*, volume 1238 of *CEUR Workshop Proceedings*, pages 51–64, 2014.
10. S. Govaerts, K. Verbert, J. Klerkx, and E. Duval. Visualizing activities for self-reflection and awareness. In *Advances in Web-Based Learning (ICWL 2010)*, volume 6483 of *Lecture Notes in Computer Science*, pages 91–100. 2010.
11. J. Itten. *Kunst der Farbe.* Otto Maier Verlag, Ravensburg, Germany, 1971.
12. J. Kay. Lifelong learner modeling for lifelong personalized pervasive learning. *IEEE Transactions on Learning Technologies*, 1(4):215–228, Oct. 2008.
13. J. Kay and B. Kummerfeld. Bringing together sensor data, user goals and long term knowledge to sup port sisyphean tasks. In *Workshop on Hybrid Pervasive/Digital Inference (HPDI 2011), Colocated with Pervasive 2011*, 2011.
14. J. Kay, N. Maisonneuve, K. Yacef, and P. Reimann. The big five and visualisations of team work activity. *Intelligent tutoring systems*, 2006.
15. D. Leony, A. Pardo, L. de la Fuente Valentín, D. S. de Castro, and C. D. Kloos. Glass: A learning analytics visualization tool. In *International Conference on Learning Analytics and Knowledge*, LAK '12, pages 162–163. ACM, 2012.
16. X. Ochoa, D. Suthers, K. Verbert, and E. Duval. Analysis and reflections on the third learning analytics and knowledge conference (lak 2013). *Journal of Learning Analytics*, 1(2):5–22, 2014.
17. V. Pammer and M. Bratic. Surprise, surprise: Activity log based time analytics for time management. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 211–216. ACM, 2013.
18. A. Pardo and C. D. Kloos. Stepping out of the box: Towards analytics outside the learning management system. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, LAK '11, pages 163–167. ACM, 2011.
19. J. A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.
20. J. Santos, K. Verbert, S. Govaerts, and E. Duval. Visualizing PLE usage. *EFE-PLE11 Workshop on Exploring the Fitness and Evolvability of Personal Learning Environments*, pages 34–38, 2011.
21. J. L. Santos, S. Govaerts, K. Verbert, and E. Duval. Goal-oriented visualizations of activity tracking: A case study with engineering students. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 143–152. ACM, 2012.
22. J. L. Santos, K. Verbert, S. Govaerts, and E. Duval. Addressing learner issues with stepup!: An evaluation. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, LAK '13, pages 14–22. ACM, 2013.
23. G. Siemens and P. Long. Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review*, 46(5):30–32+, Sept. 2011.
24. L. M. Tang and J. Kay. Lifelong user modeling and meta-cognitive scaffolding: Support self monitoring of long term goals. In *UMAP Workshops'13*, 2013.
25. F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, Nov. 2007.