Using an Elastic Stack as a Base for Logging and Evaluation of Public Displays

Christopher Rohde
christopher.rohde@unibw.de
University of the Bundeswehr Munich
Munich, Germany

Michael Koch
michael.koch@unibw.de
University of the Bundeswehr Munich
Munich, Germany

Laura Stojko
laura.stojko@unibw.de
University of the Bundeswehr Munich
Munich, Germany

ABSTRACT

This paper describes the usage of the Elastic Stack as a powerful and flexible base for capturing, storing, analyzing, and visualizing log data, as well as conducting evaluations of public displays. Nowadays, public displays are an essential element in providing information from a variety of applications and systems. As 24/7 in-person observations are impossible, support is needed for evaluation of the usage for single unit installations as well as large-scale public display networks. A software-supported solution can automate the most common data collection steps and provide a common base for further evaluation. In this paper, we present an ongoing case study on our technical solution for supporting long-term 24/7 evaluation with the Elastic Stack in any kind of public display installation. The solution is based on a logging framework and the Elasticsearch, Logstash, and Kibana – tool stack. We present the technical solution, the requirements for our implementation, and possible evaluation approaches.

KEYWORDS

Elasticsearch, Logging, Evaluation, (Semi-)Public Display, Framework, Information Radiator, Display Network

1 INTRODUCTION

Public displays have emerged into everybody's life providing easy access to any kind of information like daily news, directions, timetables, or revenue-generating advertisements. They are integrated into public spaces, train stations, airports, shopping windows as well as malls, equipped with an increasing number of sensors, such as cameras, which enable the collection of information of the targeted audience, about the content as well as installation environment parameters. To summarize, these public displays can generate a substantial amount of log data, which needs to be effectively captured, stored, analyzed, and visualized for monitoring, troubleshooting, and performance evaluation purposes.

For handling this kind of log data in a scalable and efficient manner, the Elastic Stack ([4] formerly: ELK stack; Elasticsearch, Logstash, and Kibana) has emerged as a popular toolset in recent years. For our context it was chosen based on the criteria outlined in Section 2. The Elastic Stack combines three powerful open-source tools: Elasticsearch, a distributed search and analytics engine; Logstash, a versatile data processing pipeline; and Kibana, a data visualization platform. By integrating these components, the Elastic Stack provides a comprehensive log management and analysis solution. It enables operators or organizations to centralize their log data, extract meaningful insights, and derive actionable intelligence. To integrate the Elastic Stack into the infrastructure of our long-term public display installations in the CommunityMirror project ([19], https://www.communitymirrors.net), we have developed a logging framework that helps with the objectives within our research projects and deployments (see for example [9, 19, 21]). This framework collects log data from various installations deployed in a real-world environment. We analyze the log data using Elasticsearch's robust search capabilities, leverage Logstash for data transformation and ingestion, and utilize Kibana to create intuitive visualizations, dashboards, and operations support for ongoing research activities.

In this paper, we will describe our solution as well as the requirements, the tools, and the frameworks that we use in this context. Therefore, in the first section we summarize identified challenges and requirements for our deployment, followed by a short comparison of logging toolsets and related work. Consequently, our solution's implementation and deployment are described and discussed. Finally, we present possible scenarios for which this logging approach can be beneficial.

2 CHALLENGES AND REQUIREMENTS

We already collected some experience and thoughts on what information to collect in long-term evaluations of (semi-)public display deployments – and what to use this information for [9]. Summarizing the needs from evaluation and from administration, we came up with the following information classes for a logging framework:

(1) Direct interaction (touch events, drag events, resize events, rotate events) – classical information like screen coordinates – information about (information) objects that have been touched. – Class: user-activity
(2) Activities of users in front of the screen that are not obviously, actively interacting with the screen. Information about the time period the user was present, about the interaction zone the users were in and if possible about where the user looked at. – Class: user-passive
(3) Information about what has been displayed on the screen or removed from the screen when people have interacted with it (to reproduce which objects have been on the screen at a particular time) – could be implemented by screen images

Veröffentlicht durch die Gesellschaft für Informatik e.V. in P. Fröhlich & V. Cobus (Hrsg.): Mensch und Computer 2023 – Workshopband, 03.-06. September 2023, Rapperswil (SG)

https://doi.org/10.18420/muc2023-mci-ws13-303

(in interaction relevant times). Classes: display-snapshot, display-show, display-hide.
(4) Information about what is happening in the surroundings of the screen – how many people are standing there, where they are (interaction zones [5, 16, 20, 24]), how people are moving. Classes: surroundings-snapshot, surroundings- arrive, surroundings-leave
(5) Information about source data that is available to be shown on the screen (for governance). Class: data-available, data-loaded
(6) Information about the direct environment of a screen (special events nearby, weather conditions, ...). Class: environment
(7) Information about the extended environment / context (opening hours of the building, holidays, outages of the hardware, events taking place near the screen) – this also might include free text comments by the administrators. This information class also can be used to record results of interviews or user questionnaires. – Class: comment

Log data entries can relate to a point in time or to a period. Therefore, it is important to log data always in combination with a timestamp.

In addition to simple log data entries, it would be beneficial to aggregate log data entries – e.g., to user sessions – and to make the information on the sessions available for evaluation (or administration).

A logging framework should be able to store all this information even from multiple different public display instances and allow for interactive browsing in this date – ideally in real time.

Most research on public displays is using specifically tailored tools to collect the data they need for their research questions, if not even focusing only on using manual evaluation techniques for qualitative evaluation. At least no other research can be found mentioning using a standardized log management and analytics software approach with public displays, expressing a research gap within this area for quantitative evaluation support. The importance of quantitative data collection in general was evaluated by Schwarzer et al. [26], stating the overall helpfulness of quantitative data to compare and sample quantitative interaction data to unveil latent pattern.

If implemented, logging solutions currently are customized on specific needs. Mäkelä et al [14] implement a custom logging, based on the initial ideas for a logging framework [9]. A field observation for a public display installation is evaluated with log data in a custom format that is manually entered and saved by an administrative observer [22]. More examples can be found in [6, 15, 18, 23, 25, 27]. Using a standardized log management and analytics software can make it easier to compare and sample quantitative data across varying public display installations from different institutions as it is within the nature of standardization to make comparisons and pattern recognition simpler.

There exist already several toolsets for handling (log) data like this. In general, in terms of standardizing the log data management tools, you need a tool to store the log data, a tool to gather the log data, and a tool to visualize/search the log data. So, what we were looking for was a suitable combination of these three tools, that can easily integrate and match our requirements. We examined four different toolsets, which were the most used at the time of writing. The most well known is the Elastic Stack [4]. The Elastic Stack is a popular collection of open-source tools for collecting, storing, searching, analyzing, and visualizing data from various sources in real time. It consists of the components: Elasticsearch as the storage and search engine, Logstash as the data processing pipeline and Kibana as the data exploration and visualization tool. The Elastic Stack has a big developer community and can run on premise.

Other frameworks that might be used are Splunk [3], Graylog [1] and Prometheus [2]. Splunk is a widely recognized commercial logging and log analysis platform that combines log collection, indexing, searching and visualization. Due to its commercial background, the enterprise-grade platform is known for its scalability, security, and flexibility in handling large volumes of data. Graylog is another open-source log management and analysis. It provides a centralized dashboard for log data exploration and analysis of the collected and stored log data, with powerful search capabilities and real-time monitoring. Prometheus is an open-source monitoring and alerting toolkit that includes features for collecting, storing, and analyzing time-series data, including log data. It provides a pull-based data collection model, where applications expose metrics that are scraped by Prometheus for storage and analysis.

Keeping in mind that this toolset should run on premise, due to the data privacy requirements within our projects, Splunk was not an option. Prometheus is a strong tool regarding metrics, but since we needed to support additional, varying log data formats, after evaluating Prometheus was not an option either. Graylog and the Elastic Stack both suited very well for our needs, but in terms of community and long-term support, the Elastic Stack seemed to be the better choice for us. A good indicator about community support and worldwide usage for software tools is a google trends query.

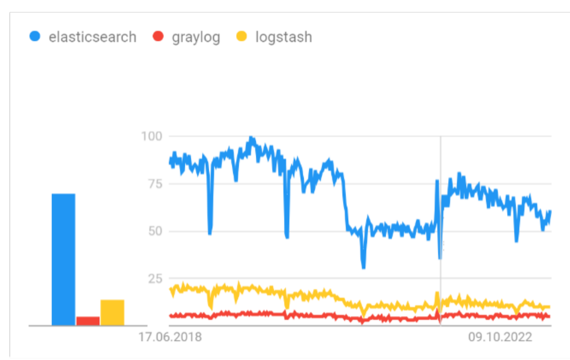

Figure 1: Google Trends Web search for toolset keywords in the past five years.

Figure 1 shows the google search queries for the mentioned keywords in relation to each other of the past five years. This indicates a wider user base for the Elastic Stack. A prototyping approach proved this; all upcoming problems could easily be solved by the detailed tool documentation and the strong community. Confirming our own experience when trying out Graylog and the Elastic Stack, the Elastic Stack has proven to be more performance optimized than Graylog too [29]. Afterall, we decided to go with the Elastic Stack as our base toolset.

3 DESIGN AND IMPLEMENTATION OF THE ELASTIC STACK INTEGRATION

In this section we present our solution for an Elastic Stack integration into the CommunityMirror-Framework, the framework used at our projects of public display installations for various research purposes (multi-user functionality [13, 17], awareness [12], as ubiquitous natural user interface (UI) [19], supporting senior's outdoor activities [7], usability for intercultural user groups [28], etc.). We will show the resulting architecture based on our requirements, how we collect and transfer the log data and what further scenarios can be covered by using the collected data.

3.1 Architecture Design

Our architecture consists of loosely coupled nodes, each responsible for a specific task, to reach a very high separation of concern as a best practice in software architecture:

- Public Display installation with a local Filebeat
- Logstash
- Elasticsearch
- Kibana

This setup gives high scalability, failure tolerance and flexibility regarding deployments when needed. In development stage, when simplicity is more important than scalability, all nodes can run on the same physical machine. This ensures short development iterations and keeps the entry level for new colleagues at a minimum.

Figure 2 shows an exemplary setup as described above. For further evaluation, the data can be accessed either by directly querying the Elasticsearch or through the Kibana user interface.

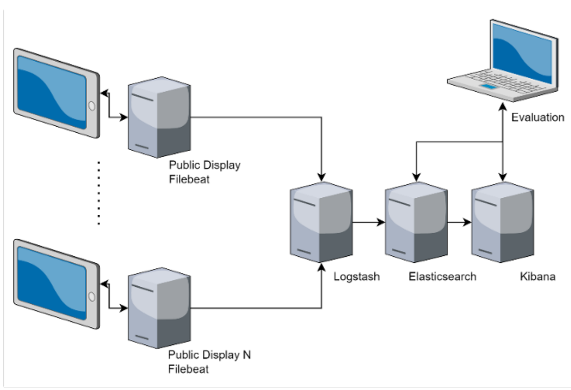

Figure 2: Elastic Stack integration architecture with multiple public displays integrated and monitored within a network.

3.2 Log Data Collection

We developed a logging framework to integrate the Elastic Stack with our existing CommunityMirror-Framework. This logging framework can be used to log data at the desired "event points" within any public display application and comes along with a predefined data model for the log data (outlined in Figure 3) in the context of public display installations based on the previous defined data collection challenges. This ensures interoperability and exchangeability between different research or usage scenarios.

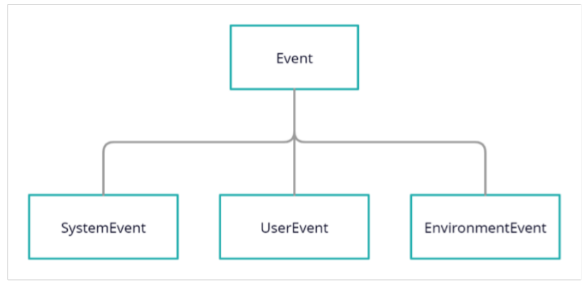

Figure 3: Data model of the logging framework.

The logging framework easily adapts to different use case scenarios, as the data model is flexible and easy to extend by introducing new event classes or action enumeration types (see Table 1). We started with an implementation of different kind of event classes categorized by the source of the event (e. g. System, User, Environment) enriched by additional event data depending on the specific type of the event.

In our current research installation [10], we operate four public displays, two additionally attached with a camera for capturing skeletal data of the people in front of the screen. To collect and persist the log data, the logging framework is used within every single public display installation and writes the uniformed data into a regular log file at the local machine. The camera log data is written into a local file as well. Log rotation guarantees that the local storage will not be exceeded. We chose not to push the log data actively into the Elastic Stack directly, as this would have a bad impact on the running public display software process performance in case any of the Elastic Stack nodes would not be reachable at any time. Instead, the earlier mentioned tool Filebeat takes this task separately. With little to no performance impact, Filebeat checks periodically the log files locally for changes and pushes any changes to a Logstash instance. Logstash than unifies the log messages again and finally pushes the messages into Elasticsearch. The data is now persisted and ready to be used for further analysis and evaluation.

3.3 Analysis, Visualization and Evaluation

With all the data in one central place, querying the Elasticsearch API (e.g. from Python scripts) or using the Kibana user interface helps to identify trends or answer research questions using the collected quantitative data in a more automated, machine-driven way. For easy-to-use analysis and evaluation, especially Kibana offers a wide range of tools to discover patterns and usage effects.

With Kibana visualizations of the log data can be easily configured by selecting the desired data fields, applying aggregations and defining the data granularity. Figure 4 shows a custom diagram filtered for user interactions within a one-month period, helping to identify usage peaks as shown near the end of displayed period. With this kind of diagrams, researchers can more easily get aware of time period of specific interest.

Aggregations such as count, sum, average and unique count can be applied to the data to derive even more meaningful metrics.

Name	Description	JSON-Key
timestamp	Date-Time when the event occurred	DATE
instanceId	ID of the application instance	INSTANCEID
eventSource	Source of the event; enumeration	SOURCE
eventType	Type of the event, e. g. DISPLAY_SHOW for the entry of an element; enumeration	TYPE
eventData	Additional data within the context of the event, e. g. the position of a source event or the type of user activity	DATA
usersession	ID for user session identification	SESSION

Table 1: Event-Datatype of the developed logging framework data model

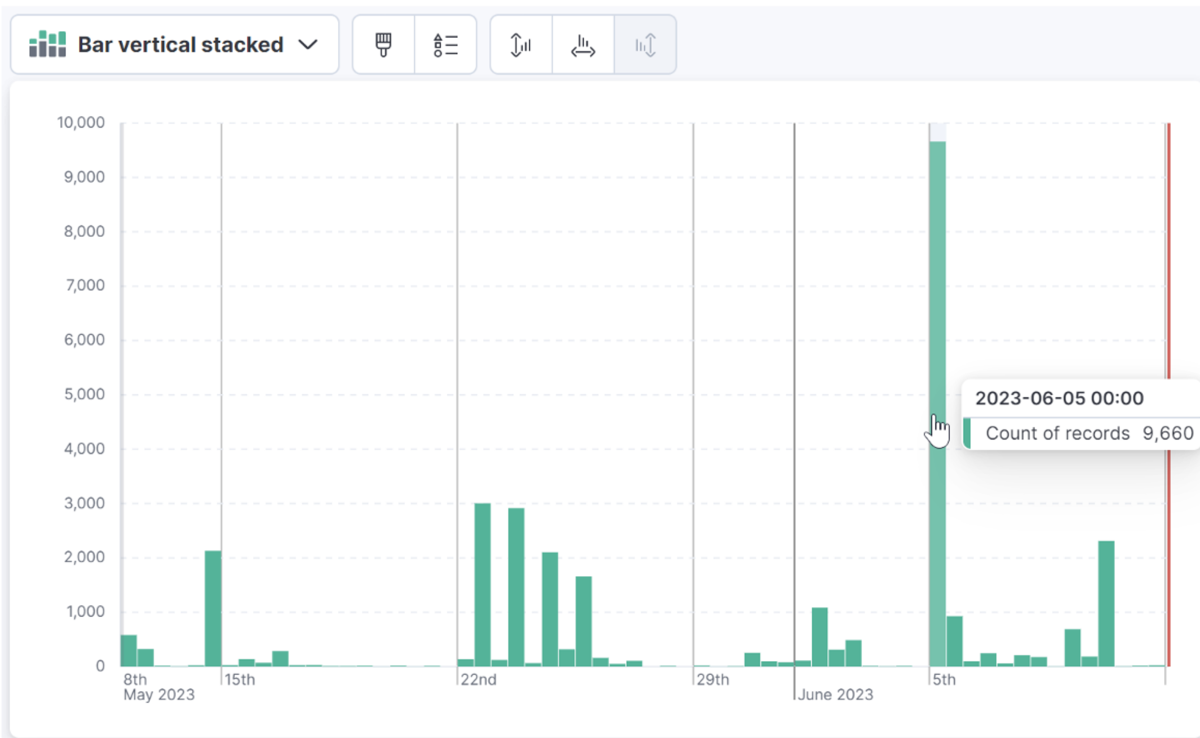

Figure 4: Example of a Kibana diagram showing the user interaction count per day for one CommunityMirror.

To summarize and arrange multiple visualization into a common, reusable frame, Kibana also supports the creation of dashboards. Dashboards allow users to arrange and organize multiple visualizations on a single page. Users can even interact with the visualizations within the dashboard, applying filters and drilldowns to explore specific aspects of the log data within a specified time frame. Figure 5 shows a dashboard visualizing the user interactions and the count of error messages. Given a peak within the error messages, inspecting the detailed error messages the cause can be easily identified.

For automatically identifying such issues, Kibana facilitates real-time monitoring of log data through its alerting and notification mechanisms helping to operate the public display installations remotely. Alert conditions can be defined based on log data metrics or specific events such as a predefined amount of error messages within a fixed time frame and set up alert actions, for example sending email notifications or triggering external systems. This feature enables proactive monitoring and immediate response to critical events or anomalies detected in the log data and helps to easily monitor running public display installations. Within the CommunityMirror project, we use this monitoring and alerting functionality to identify installation problems, downtimes or other kind of failures, where we prior had to manually search the log data or visit the installations in person.

3.4 A Usage Example

To show how the solution can be used in context we briefly describe one example from our current work:

In the CommunityMirror Network [10] we are currently operating four large screens as interactive information displays. All the log data from the screens is directly forwarded to our Elastic Stack

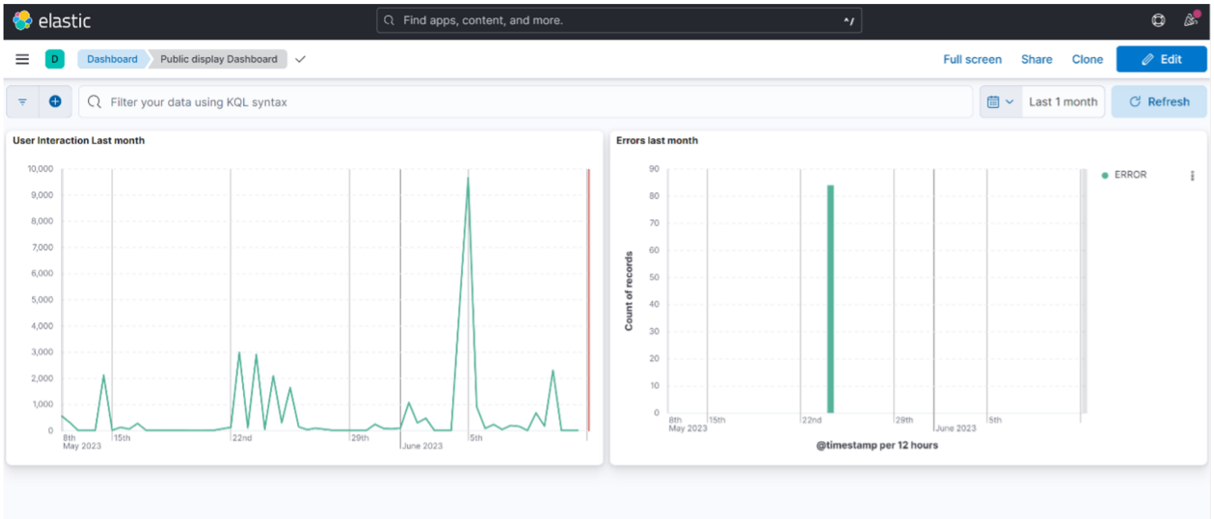

Figure 5: Dashboard of a public display network.

installation beginning 1st March 2023 - including events for every TOUCH and every DRAG interaction of the users with the screens.

A quite common task is to check if a new functionality on the screen influences the usage of the screen. In our example the functionality was a light gamification approach (adding a basketball basket to throw info particles into it).

For evaluating the functionality we planned to compare an average count of user activities on the screen between periods with the gamification solution to periods without the gamification solution.

So, we first just fixed a two week period in which the functionality should be activated. After the experiment has finished, we started exploring what has happened. Using Kibana we easily could focus on the one screen that was changed and on the two events indicating user activity (touch events and drag events). Looking at an interval of several weeks before the experiment started, we could see quite regular usage except on weekends and on holidays. There were two days with exceptionally high usage (+1000%). Looking more deeply into these days it showed that the high usage was due to technical problems and a demo tour of the installations. We decided to exclude the weekends, holidays and the identified peaks from the analysis. For the experimental interval we did the same exploration in Kibana. The exploration showed a much higher usage (>=+200%) in the first 2-3 days after the beginning that vanished afterwards. We decided to analyse these days separately since it seemed to be a Novelty Effect [11]. In the meantime the installation continued to collect data, so we also had the chance to look into the data from after the experiment was stopped.

We decided to export the data from six weeks before the experiment to four weeks after the experiment excluding weekends, holidays and peaks with >1000% - using a Python script that was quickly programmed. The exported data then was imported into Excel for further analysis and visualization.

4 SUMMARY AND OUTLOOK

In this paper we explored the usage of the Elastic Stack for logging and evaluations of public display installations. We began by a short definition of public displays followed by brief overview about the challenges and requirements associated with logging and evaluation in public displays. The data collection, real-time nature of data generation, diverse data formats, scalability and performance considerations and log data retention and management were among the key challenges and requirements identified. After evaluating different logging approaches, we then introduced the Elastic Stack, which served as the core toolset of our solution. We delved into the design and implementation of the resulting Elastic Stack integration, which data we collect and the resulting analysis and evaluation possibilities.

In conclusion the Elastic Stack as the core of our implementation approach presents a robust and efficient solution for logging and evaluations of public display installations. There might be other toolsets, that also fulfil the requirements, but by leveraging the capabilities of Elasticsearch, Logstash, and Kibana, we can effectively capture, analyze, and visualize log data from public displays, enabling proactive monitoring, troubleshooting, and performance optimization as well as a more automated approach to answer current or upcoming research questions more efficiently. The design and implementation considerations discussed in this paper provide a foundation for successfully implementing the Elastic Stack integration in various environments and reaping the benefits of comprehensive log data analysis for many upcoming evaluation tasks.

We currently have mainly addressed system and user activity events – and have omitted user passive events. These would have to be derived from the skeletal data captured by the camera sensors. However, the current framework for capturing skeletal data [8] already allows for direct processing or batch processing this information for user passive events e.g. when a user enters a particular interaction zone in front of the screen. This will be added in the near future.

In the future we will focus on validating the benefits of our logging framework in our long-term deployments by trying automatic evaluation approaches of the deployment usages and effects along with further extensions and enhancements of the introduced data model to support an additional, more standardized evaluation approach for long-term deployments of public displays. As we research large semi-public displays from multiple angles, this logging framework allows quantitative evaluations to be conducted 24/7 without the need for personal on-site involvement, making our research easier and more scalable, and even enabling new research perspectives on the long-term deployments of display networks.

REFERENCES

[1] 2023. Graylog. https://www.graylog.org/
[2] 2023. Prometheus. https://prometheus.io/
[3] 2023. Splunk. https://www.splunk.com/
[4] 2023. The Elastic Stack. https://www.elastic.co/elastic-stack/
[5] Florian Alt, Andreas Bulling, Lukas Mecke, and Daniel Buschek. 2016. Attention, please! Comparing Features for Measuring Audience Attention Towards Pervasive Displays. In *Proc. Designing Interactive Systems (DIS)*.
[6] Florian Alt, Stefan Schneegaß, Albrecht Schmidt, Jörg Müller, and Nemanja Memarovic. 2012. How to evaluate public displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays (PerDis 2012)*. ACM Press, New York, New York, USA, 1–6. https://doi.org/10.1145/2307798.2307815
[7] Julian Fietkau. 2023. *Gestaltung von Quests in vernetzten Informationsstrahlern im urbanen Raum zur Steigerung der Teilhabe älterer Menschen.* Dissertation. Universität der Bundeswehr München, Neubiberg, Germany. https://doi.org/10.18726/2023_1
[8] Julian Fietkau. 2023. A New Software Toolset for Recording and Viewing Body Tracking Data. In *Mensch und Computer 2023 – Workshopband*, Peter Fröhlich and Vanessa Cobus (Eds.). Gesellschaft für Informatik e.V., Bonn, Germany, 4 pages. https://doi.org/10.18420/muc2023-mci-ws13-334
[9] Michael Koch. 2019. Towards a Logging Framework for Evaluation and Management of Information Radiators. In *Mensch und Computer 2019 – Workshopband.* Gesellschaft für Informatik e.V. https://doi.org/10.18420/muc2019-ws-566
[10] Michael Koch, Julian Fietkau, and Laura Stojko. 2023. Setting up a long-term evaluation environment for interactive semi-public information displays. In *Mensch und Computer 2023 – Workshopband*, Peter Fröhlich and Vanessa Cobus (Eds.). Gesellschaft für Informatik e.V., Bonn, Germany, 5 pages. https://doi.org/10.18420/muc2023-mci-ws13-356
[11] Michael Koch, Kai von Luck, Jan Schwarzer, and Susanne Draheim. 2018. The Novelty Effect in Large Display Deployments – Experiences and Lessons-Learned for Evaluating Prototypes. In *Proc. 16th European Conference on Computer-Supported Cooperative Work*. European Society for Socially Embedded Technologies (EUSSET). https://doi.org/10.18420/ecscw2018_3
[12] Stefan Lippert. 2020. *Nutzen aus der Perspektive von Awareness bei Informationsstrahlern als zusätzliche sozio-technische Systeme in organisationalen Gruppen.* Dissertation. Universität der Bundeswehr München.
[13] Eva Lösch. 2020. *Unterstützung der Exploration von mehrbenutzerfähigen interaktiven Informationstafeln im (halb) öffentlichen Raum.* Dissertation. Universität der Bundeswehr München. https://athene-forschung.unibw.de/131737
[14] Ville Mäkelä, Jonas Winter, Jasmin Schwab, Michael Koch, and Florian Alt. 2022. Pandemic Displays: Considering Hygiene on Public Touchscreens in the Post-Pandemic Era. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* Association for Computing Machinery. https://doi.org/10.1145/3491102.3501937
[15] Nemanja Memarovic, Ivan Elhart, and Elisa Rubegni. 2016. Developing a Networked Public Display System. *IEEE Pervasive Computing* 15, 3 (2016), 32–39. https://doi.org/10.1109/MPRV.2016.59
[16] Jörg Müller, Florian Alt, Daniel Michelis, and Albrecht Schmidt. 2010. Requirements and design space for interactive public displays. In *Proceedings of the international conference on Multimedia - MM '10.* ACM Press, 1285–1294. https://doi.org/10.1145/1873951.1874203
[17] Andrea Nutsi. 2018. *Gestaltungsempfehlungen für mehrbenutzerfähige Informationsanwendungen auf interaktiven Wandbildschirmen im (halb-)öffentlichen Raum.* Dissertation. Universität der Bundeswehr München. https://athene-forschung.unibw.de/126792
[18] Timo Ojala, Vassilis Kostakos, Hannu Kukka, Tommi Heikkinen, Tomas Linden, Marko Jurmu, Simo Hosio, Fabio Kruger, and Daniele Zanni. 2012. Multipurpose Interactive Public Displays in the Wild: Three Years Later. *Computer* 45, 5 (may 2012), 42–49. https://doi.org/10.1109/MC.2012.115
[19] Florian Ott. 2018. *CommunityMirrors: Interaktive Großbildschirme als ubiquitäre Natural User Interfaces für Kooperationssysteme.* Dissertationsschrift. Universität der Bundeswehr München. https://doi.org/10.18726/2018_1
[20] Florian Ott and Michael Koch. 2012. Social Software Beyond the Desktop — Ambient Awareness and Ubiquitous Activity Streaming. *it - Information Technology* 54, 5 (2012), 243–252. https://doi.org/10.1524/itit.2012.0687
[21] Florian Ott and Michael Koch. 2019. Exploring Interactive Information Radiators – A Longitudinal Real-World Case Study. In *Mensch und Computer 2019 – Workshopband.* Gesellschaft für Informatik e.V. https://doi.org/10.18420/muc2019-ws-565
[22] Callum Parker, Martin Tomitsch, and Judy Kay. 2018. Does the Public Still Look at Public Displays? A Field Observation of Public Displays in the Wild. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 2. 73:1—-73:24. https://doi.org/10.1145/3214276
[23] Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. 2008. "It's Mine, Don't Touch!": Interactions at a Large Multi-Touch Display in a City Centre. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'2008)* (2008), 1285–1294. https://doi.org/10.1145/1357054.1357255
[24] Thorsten Prante, Carsten Röcker, Norbert Streitz, and Richard Stenzel. 2003. Hello.Wall – Beyond Ambient Displays. In *Adjunct Proceedings of the 5th International Conference on Ubiquitous Computing (UBICOMP'03).* Seattle, 277–278. https://doi.org/10.1.1.58.3459
[25] Yvonne Rogers, William R Hazlewood, Paul Marshall, Nick Dalton, and Susanna Hertrich. 2010. Ambient influence: can twinkly lights lure and abstract representations trigger behavioral change? *Proceedings of the 12th ACM international conference on Ubiquitous computing* (2010), 261–270. https://doi.org/10.1145/1864349.1864372
[26] Jan Schwarzer, Kai von Luck, Susanne Draheim, and Michael Koch. 2019. Towards Methodological Guidance for Longitudinal Ambient Display In Situ Research. In *Proc. Europ. Conf. on Computer-Supported Cooperative Work*. EUSSET. https://doi.org/10.18420/ecscw2019_ep07
[27] Ben Shelton and Keith Nesbitt. 2017. Evaluating WaveWatch: an ambient display of web traffic. In *Proceedings of the Australasian Computer Science Week Multiconference, ACSW 2017, Geelong, Australia, January 31 - February 3, 2017.* 9:1—-9:9. https://doi.org/10.1145/3014812.3014821
[28] Laura Stojko. 2022. Cultural Design Compass: Who Do You Design For?. In *Proceedings of the 20th European Conference on Computer-Supported Cooperative Work: The International Venue on Practice-centred Computing on the Design of Cooperation Technologies.* https://doi.org/10.48340/ecscw2022_p09
[29] Roxana-Daniela Şuştic, Alexandra Moraru, Andrei-Bogdan Rus, and Virgil Dobrota. 2022. Performance evaluation of ELK stack versus Graylog as Open-Source log management tools. *Acta Technica Napocensis: Electronica - Telecomunicatii; Cluj-Napoc* 62, 1 (2022), 17–22.