DEMO

Interaction Visualization and Analysis in Automation Industry

Christian Grossauer
University of Applied Sciences
Upper Austria
Department of Mobile Computing
Softwarepark 11, 4232
Hagenberg, Austria
christian.grossauer@fh-hagenberg.at

Dustin Steiner
University of Applied Sciences
Upper Austria
Department of Mobile Computing
Softwarepark 11, 4232
Hagenberg, Austria
clemens.holzmann@fh-hagenberg.at

Clemens Holzmann
University of Applied Sciences
Upper Austria
Department of Mobile Computing
Softwarepark 11, 4232
Hagenberg, Austria
dustin.steiner@fh-hagenberg.at

Andreas Guetz
University of Applied Sciences
Upper Austria
Department of Mobile Computing
Softwarepark 11, 4232
Hagenberg, Austria
andreas.guetz@fh-hagenberg.at

MUM '15 November 30 - December 02, 2015, Linz, Austria

ACM ISBN 978-1-4503-3605-5/15/11.

DOI: http://dx.doi.org/10.1145/2836041.2841218

Abstract
For a mobile developer, being able to see how users interact with a mobile application represents a valuable additional means for improving the user experience. In the automation industry, applications that are easy to use lead to less frustration and therefore shorter downtime of the machines. In this paper, we propose a visualization prototype and some general recommendations for visualizing and analyzing user interaction data of the automation industry.

CCS Concepts
•**Human-centered computing** → **Visual analytics; Visualization toolkits;** Ubiquitous and mobile computing design and evaluation methods;

Author Keywords
Interaction analysis; Automation industry; Sankey diagram

Introduction
Understanding how people use an application plays an important role for developers of all kinds of sectors. In this respect, the automation industry is not different. Humans interact with computers (here often machines) using either physical controls (e.g. hardware buttons and knobs) or, more and more frequently, digital control elements (e.g. virtual buttons and any other control elements on digital user interfaces). The developers of those human-machine-

Terminology

Mask: A mask is a visual representation of screen content. Visiting a mask means the mask is opened and stays open for a while, interacting with the mask means any kind of actions are executed on the mask.

Sequence: A series of mask visits is called a sequence. A sequence can be seen as everything that happens after the login until the user is logged out (manual or automatic time out).

Usage metrics: For every mask visit, we collected the time spent on the mask (dwell time), the number of interactions executed on the mask, the executing user (user ID), the current operating status of the machine and the time of the day (timestamp).

interfaces create specific designs and interaction concepts, defining the visual appearence of the user interfaces and how users can and should use them. The functionality of quality software is tested extensively, therefore errors/exceptions/bugs are rare appearances. But even if there are no functional problems with the graphical user interface and the logic behind it, there still might be usability issues. If users have problems finding a certain functionality or have to execute badly designed tasks over and over again, they get frustrated. Fortunately, usability/user-experience tests, where these kind of problems can be recognized in advance, are getting more and more popular.

In this paper, we present visualization and analysis possibilities for interaction data in the automation industry. Based on a requirements analysis we conducted in 2013 [7], we created prototypes for visualizing and analyzing interaction data. Based on a first project with a company from the automation industry [3], we created visualization prototypes and tested them with another large enterprise. These new visualization prototypes, as well as our findings from creating them, will be described in the following.

Motivation
Conducting usability tests (especially field studies) is often expensive and time consuming. Therefore, gathering interaction data in the wild would be a first step in reducing the efforts. Visualizing this collected data in a comfortable way would be the second step, which would help to save a lot of time and money.

This is of utmost importance in the automation industry, where task execution time is directly related to revenue: A machine produces a certain amount of items per hour, and less if there are problems in the user interface requiring longer or even unsuccessful interactions with the machine. Therefore, optimizing the responsible applications and workflows has a direct impact on the production output.

Based on the results of the requirements analysis, we focused on how users navigate through an application. Additionally to what masks the users visited, we analyzed dwell times, the number of interactions and some contextual data like user ID, operating status and time of the day.

Related Work
Many analytic tools offer analysis possibilities for app navigation, page visits, session times and even time of the day (e.g. Flurry[1], Localytics[2], Mixpanel[3], TestFlight[4] or Google Analytics[5]) next to other “traditional” metrics like loyalty, conversions and location information. Unfortunately, none of the considered tools could fulfil the requirements of the automation industry for presenting the available interaction data in a way that makes it possible to gain some additional value out of it. Therefore we started to create customized visualizations, which are based on Sankey diagrams. Sankey diagrams are traditionally used for visualizing a flow of energy or material in networks and processes [6]. An interactive Sankey diagram was used by Riehmann et al. [5] to explore complex energy flows, therefore we decided to create a Sankey based visualization to visualize the navigation flow through an application. This visualization is tailored for interaction data from the automation industry.

Interaction Data
To visualize how people use an application, we first needed to track the appropriate interaction data. While our AUTo-

[1] https://developer.yahoo.com/flurry/
[2] http://www.localytics.com/
[3] https://mixpanel.com/
[4] https://developer.apple.com/testflight
[5] http://www.google.com/analytics/

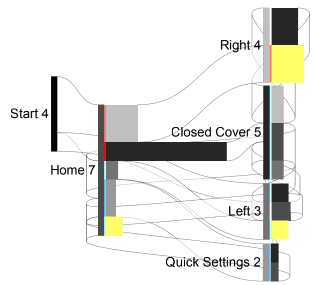

Figure 1: Example of navigation paths in Overview Mode. The height represents the number of mask visits, the width of the horizontal bars displays additional informations (dwell time or number of interactions).

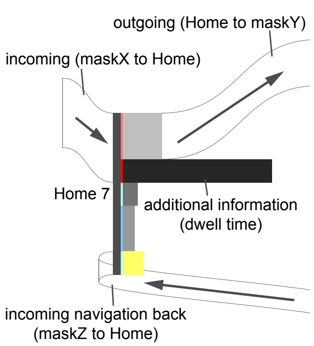

Figure 2: Simplified details for one mask in Overview Mode. Incoming and outgoing paths indicate the navigation flow. The horizontal bars are filled with the same color as the particular target mask, yellow represents the end of a sequence.

MAte framework was used to collect the needed interaction data in the first project [3], the second company was not able to use our framework due to their different operating platform, but provided us with the needed data by using their own data collection tools.

The data collection was focused on how the users navigate through the application, by tracking the visited masks, the time spent and the number of interactions happened on each mask. We call a chronological list of the visited masks and the associated additional data (time and number of interactions) a sequence (see side bar on page 2 for terminology).

Visualization Prototype

Based on our previous work [4], we created a new prototype tailored for the needs of the automation industry.

Navigation cycles and additional information
Sankey diagrams often do not visualize cyclic flows. In case of a cycle (navigation back), we simply added paths leading back to already visited masks. Afterwards, we started to incorporate additional information. First, we embedded only dwell times per mask, by adding horizontal bars to each mask, where the length of the bar indicates the value. After that, we added a possibility to switch between dwell times and number of interactions, which are visualized in the same way. This enables to see how much time was spent or how many interactions happened on each mask and makes it easily comparable across all masks (see figure 1 and 2).

Colors and paths
Working with the first iteration of the visualization prototype, we encountered some problems and came across some interesting new features. One of the first problems was the (compared to the mobile world) bigger amount of different masks. While there is no specific value for the average number of mobile app "masks", the average number of masks in our cases (from the automation industry) was around 50. The large amount of masks became a problem for our initial color coding, which was based on d3's categorical colors category20[6]. We thought of simply increasing the number of distinct colors, by providing variations of each of the existing color of d3's color palette. But there were either (a) too few distinct colors or (b) too many colors (a "color overload"). Harrower et al. [1] described a similar problem when coloring maps. Because we do not know how many different colors we need, we decided to use only grey tones. This limits the maximum number of different colors for masks to 255.

Another problem with the huge number of different masks was occlusion of connecting paths between the masks. With a handful of masks, the occlusion was no big deal, but when dealing with a larger number, a lot of paths could not be followed correctly. Therefore we added the possibilities to either show/hide all the paths or toggle visibility by hovering over the corresponding mask. Another possible solution would be a variation of edge bundling as described by Holten [2].

Overview and Expanded Mode
During the first tests with interaction datasets from the automation industry, we identified a problem resulting from the large number of individual masks. From our first project as well as from our experience in the mobile app development, we were used to a lower number of masks, therefore our first visualization prototype was not optimized for the large number. We came up with two solutions to compensate that problem. First, providing a set of filter possibilities

[6] https://github.com/mbostock/d3/wiki/Ordinal-Scales#category20

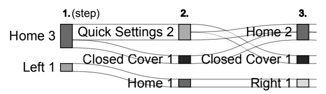

Figure 3: Example of navigation paths in Expanded Mode. Here, only the first three interactions of all sequences are shown.

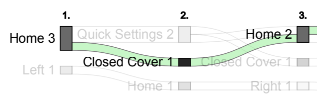

Figure 4: One specific navigation path (Home, Closed Cover, Home) highlighted in Expanded Mode.

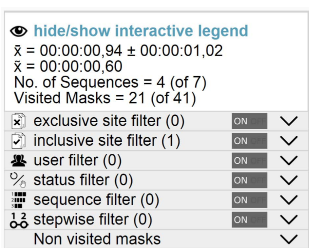

Figure 5: Interactive legend with general information (average dwell time, no. of sequences and visited masks) and filter possibilities.

(see following section) to limit the number of masks and second, a different mode to visualize the navigation. We called the first visualization mode *Overview Mode*, because it allows the user to see the overall navigation. All visits of a certain mask are aggregated over the whole dataset, therefore common mask visits can easily be identified by the height of the corresponding node (see figure 1). This mode works well to get an overview of how often users visited each mask, but lacks in the possibility to see the exact order how the users visited the masks.

Figure 3 shows a part of the same dataset in the so called *Expanded Mode*. Here, the visited masks are displayed in the same order as they were visited. This offers the possibility to see exactly how the users navigated through the application (see figure 4), but lacks in being able to see common masks at a glance.

Filtering

Dealing with (potential) big amounts of data, filtering is very important to get useful results. The first filter possibility we provided is a time filter (e.g. show only data between t1 and t2). As we have interaction data with reoccurring time patterns (e.g. work week, 1/2/3 shifts, periodical tasks etc.), we added the possibility to filter by weekday (show only data from Mondays) and between daytime (e.g. show only data from 9:00AM to 5:00PM but from every day a week).

The second and third filter (see figure 5 for all filters except the time filter) are based on the visited (or not visited) masks. We called them *exclusive* and *inclusive* filter. Exclusive means, we exclude all sequences where the selected mask was at least visited once. Inclusive means, we include only sequences where the selected mask was at least visited once.

Based on the working environments of an automation company, we included possibilities to filter by user or operating status. This can be very helpful, because the usage (therefore the navigation flow) differs strongly between users (e.g. between a normal worker, a foreman and a service engineer) and the actual operation status (e.g. the machine runs in automatic, manual or set-up mode). Depending on these parameters, other masks are visited and different dwell times and numbers of interactions occur.

Another filter we provided is a simple sequence filter which enables to show or hide certain sequences.

The last filter we provided is again based on visited masks. It behaves like the inclusive filter, but it can take the step number of the mask visit in account (e.g. show only data where the "Home" mask was visited at first). Therefore we called it *stepwise*, because of its functionality to filter out sequences, where the selected mask was not visited at a certain step in the calling sequence.

Conclusion

Together with an automation company, we used the created prototypes to analyze several interaction datasets and came across interesting findings. We encountered masks with unusual dwell times or numbers of interactions as well as recurring navigation patterns, which formed a good starting basis for further analysis. We could derive the preferred ways how users trigger the execution of certain tasks. The Expanded Mode helped recreating scenarios when users opened certain masks (e.g. what was done before help was visited). Additionally, we came across an error in the allocation of user rights (a user had unauthorised access to certain masks). All in all, the created prototypes can provide valuable insights in how users interact with an application. Especially when dealing with a large number of masks, we recommend to provide a wide variety of filters. We suggest to provide a view showing the whole navigation at once as well as a view that is focused on individual sequences.

Acknowledgments

The research presented is conducted within the Austrian project “AUToMAte - Automatic Usability Testing of Mobile Applications” funded by the Austrian Research Promotion Agency (FFG) under contract number 839094.

References

1. Mark Harrower and Cynthia A. Brewer. 2003. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal* 40, 1 (June 2003), 27–37. DOI: http://dx.doi.org/10.1179/000870403235002042
2. Danny Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (Sept. 2006), 741–748. DOI: http://dx.doi.org/10.1109/TVCG.2006.147
3. Clemens Holzmann, Florian Lettner, Christian Grossauer, Werner Wetzlinger, Paul Latzelsperger, and Christian Augdopler. 2014. Logging and Visualization of Touch Interactions on Teach Pendants. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 619–624. DOI: http://dx.doi.org/10.1145/2628363.2628429
4. Florian Lettner, Christian Grossauer, and Clemens Holzmann. 2014. Mobile Interaction Analysis: Towards a Novel Concept for Interaction Sequence Mining. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 359–368. DOI: http://dx.doi.org/10.1145/2628363.2628384
5. P. Riehmann, M. Hanfler, and B. Froehlich. 2005. Interactive Sankey diagrams. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. 233–240. DOI: http://dx.doi.org/10.1109/INFVIS.2005.1532152
6. Mario Schmidt. 2008. The Sankey Diagram in Energy and Material Flow Management. *Journal of Industrial Ecology* 12, 1 (2008), 82–94. DOI: http://dx.doi.org/10.1111/j.1530-9290.2008.00004.x
7. Werner Wetzlinger, Dietmar Nedbal, Andreas Auinger, Christian Grossauer, Clemens Holzmann, and Florian Lettner. 2014. Mobile Usability Testing Requirements and their Implementation in the Automation Engineering Industry. In *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia (MoMM '14)*. ACM, New York, NY, USA, 62–71. DOI: http://dx.doi.org/10.1145/2684103.2684120