Usage Analysis of Cross-Device Web Applications

Maria Husmann, Nicola Marcacci Rossi, and Moira C. Norrie
Department of Computer Science
ETH Zurich, Switzerland
{husmann,norrie}@inf.ethz.ch, nicolamr@student.ethz.ch

ABSTRACT
Applications that adapt to and make use of the set of available devices have received a lot of attention lately. Even so, it remains unclear how such cross-device applications are used outside of controlled environments like lab studies where users are instructed on usage. Insights into the usage of an application could help measure business goals and improve the application, as is common for web-based applications. Building on practices from the web, we propose a system for analysing the usage of web-based cross-device applications. We introduce metrics and use cases of interest in cross-device scenarios and explain how we support these in our implementation, XD-Analytics. Finally, we discuss how we used XD-Analytics to observe the introduction of a cross-device feature in an existing application with a substantial user base.

ACM Classification Keywords
H.5.m Information interfaces and presentation (e.g., HCI): Miscellaneous

Author Keywords
cross-device; usage analysis; parallel multi-device usage; web analytics

INTRODUCTION
Devices are now ubiquitous in our lives, ranging from personal devices such as smartphones, tablets, and watches to shared ones such as smartTVs and public displays. The abundance of devices has led to situations where we have not just a single device at hand but multiple [6]. There are potential benefits in coordinating functionality among devices [18]. This has been recognised by the research community and numerous projects investigating cross-device interaction styles and applications have been proposed [9, 15, 17, 19, 8].

In particular, recent advances in web technologies have sparked the creation of many web-based cross-device frameworks [10, 24, 1, 20, 14, 7]. Despite the number of frameworks, little is known about how cross-device applications are used outside of laboratories where users are typically equipped with a specific set of devices and instructed on how to use the application. Furthermore, to the best of our knowledge, current cross-device frameworks do not collect any data or statistics on usage. Consequently, once an application has been deployed, it is hard to assess how it is actually used. While many cross-device applications are built to be flexible and support almost any combination of devices, knowing which combinations are actually used would be valuable feedback and could be used to further optimise an application. Furthermore, as cross-device applications are still rare outside of laboratories, people may not understand how they can be used and might possibly use them with only a single device. If this could be detected, effort could be made to prompt users to engage with other available devices.

While there is little support for analysis specific to cross-device applications, techniques to gather insights into the usage of regular websites and applications are well-established [21, 2]. A number of tools, both free and commercial, are available to analyse traffic to websites and the behaviour of their users. These tools track metrics such as the device, browser, and operating system used to access a website, the location of the user, the time spent on the various parts of a website, and behaviour such as the navigation paths taken. This information can then be used to tweak and improve a given website, but also to measure business goals and gain valuable insights on the users and usage of the website.

In this paper, we explore the usage analysis of cross-device web applications. First, we discuss existing work in cross-device applications and web usage analysis. We then introduce XD-Analytics, our system for analysing cross-device application usage. Finally, we discuss how we used the system to observe the introduction of cross-device functionality into an existing system with a substantial user base.

BACKGROUND
As computing devices have become pervasive, researchers have started to study how people use these devices. A number of interesting interview studies revealed usage patterns and issues in coordinating multiple devices, that provided inspiration for cross-device interactions and applications.

Kawsar et al. [13] investigated the devices that families use in their homes, using activity logs, a survey, and interviews. They observed spatial and temporal dynamics for devices and application usage. Further, they noted that the Desktop PC is turning into a niche platform (e.g. used for gaming) and

PerDis '16, June 20–12, 2016, Oulu, Finland
©ACM. ISBN 978-1-4503-4366-4/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2914920.2915017

that its role as the main computing device was being taken over by tablets and smartphones. In an early study, Dearman et al. [4] interviewed 27 persons from academia and industry, investigating why and how they use multiple devices for work and personal activities. Published in 2008, the study was executed at a time when smartphones were only starting to proliferate. Only 10 smartphones were reported among all of the participants, a number which would be expected to be much higher today. The average participant had a device collection of roughly six devices, including digital cameras. The authors noted that better synchronisation and information transfer between the devices was needed.

This notion was confirmed by a more recent interview study [18] focusing on activities in distributed workspaces. The authors observed serial and parallel usage patterns of multiple devices and also stress the need for improved functional coordination between the devices. Sequential and parallel patterns were also discovered by Jokela et al. [11] in an interview and diary study. The authors differentiate parallel patterns into resource lending (e.g. using the phone as a hotspot), unrelated parallel use (using multiple devices for multiple tasks), and related parallel use (using multiple devices for a single task).

Cross-device applications typically aim at related parallel use by distributing the user interface of a single application across multiple devices. Examples include distributed video players [24], photo galleries [20], maps [17], and slideshow presenters [16]. For the distribution of the application elements, device roles [16, 5] and characteristics [24] such as input modalities or screen size are taken into account. The screen size is also considered in a taxonomy for multi-user, multi-device ecologies by Terrenghi et. al [22]. The ecosystems are classified using the dimensions of scale and social context. For scale, the screen size as well as viewing distance and angle are taken into account.

Another methodology for analysing user behaviour is based on logging data on devices. Church et al. [3] provide a comprehensive summary of work done in the area of logging on mobile devices. Devices are required to install a logger (which can also be integrated into an application) that can track a wide range of data types including application launches, motion data, and screen state. Analysis is usually limited to a single device per user, with the exception of ENGAGE [12], which has similar goals to ours and attempts to measure engagement in multi-device settings. On desktops and laptops, cameras are used to track the user's face, while an app tracks the focused application and screen state periodically on mobiles. While such tracking approaches produce a lot of interesting data, they are rather heavyweight and can raise privacy concerns with users. They are thus mostly used for limited durations and often with smaller sets of participants (a notable exception being [23] with 12'500 participants tracked over two years).

Given that a lot of cross-device applications are built with web technologies, we propose that approaches for analysing web usage [2, 21] could be applied. These are generally more lightweight and do not require any installation on client devices. As a trade off, less data is typically logged resulting in a less comprehensive picture, but it also introduces fewer privacy issues. Furthermore, while most studies mentioned so far focus on general device usage, web tracking is best suited to analysing the usage of a single application. A range of commercial systems such as Google Analytics[1], Alexa[2], and Web-Stat[3], provide such tracking services that can be integrated into any website and then generate reports and visualisations of the tracked data. Google Analytics offers basic cross-device reports[4] that associate multiple devices with a user. This information is used to show *device overlap*, the device types and numbers thereof that a person uses to access an application. Sequential usage patterns are tracked in *device paths*, which show the 5 last devices used before a conversion, for example a purchase. Parallel usage patterns are not tracked, but these are of particular interest in cross-device scenarios and this is a gap that we attempt to fill with XD-Analytics.

XD-ANALYTICS

XD-Analytics provides insights into the usage of cross-device applications. Data is collected from client applications and presented to the analyst through a versatile user interface. Bar charts, time line charts, and ordered lists can be used to view the data. XD-Analytics provides extensive filtering options and has been built on principles from faceted search. When a value in a facet (e.g. devices) is selected, the other facets update to only reflect values that also match the selected facet (e.g. locations that were visited with the device). This gives a lot of flexibility to the analyst and they can incrementally build up charts or drill down into aspects of interest.

Metrics

In addition to the metrics commonly tracked by analytics systems such as locations, time online, and the browser, we introduce and track the following metrics.

Devices, Users, and Sessions The two most important entities recorded are devices and users. Each device is associated with a user, as soon as a user is logged in. Additionally, each open tab in a device corresponds to a session. Thus, a user can have multiple devices which can have multiple sessions.

Device Types Devices are classified into five categories (xs, sm, md, lg, xl) according to their screen size in CSS pixels[5]. While not entirely accurate for all devices due to differences among vendors, informal tests produced good results with mobile phones (Nexus 4, Samsung Galaxy Note 4) classified as small, a tablet (iPad Air) as medium, a notebook as large, and a 30inch screen as extra large.

Device Combinations When a user accesses the application with multiple devices simultaneously, the device types used are grouped into a device combination. Figure 1 illustrates this metric showing, for example, that 44 users used two large screen devices at the same time to access the application. (note that this and the following screenshots have been edited to give better readability). The next most common combination is a

[1] http://www.google.com/analytics/
[2] http://www.alexa.com/tools
[3] http://www.web-stat.com/
[4] https://support.google.com/analytics/answer/3234673
[5] https://www.w3.org/TR/css3-values/#reference-pixel

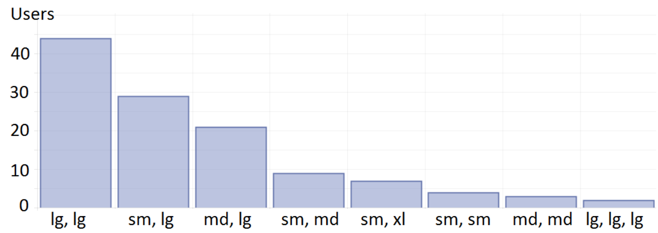

Figure 1. Device combinations

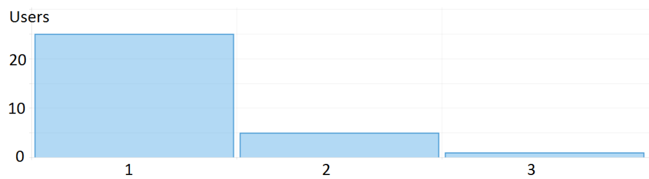

Figure 2. Device cardinalities

small and large screen. Only a single combination consists of more than two devices.

Device Cardinalities The device cardinalities count the number of devices in a combination regardless of the device type. Figure 2 displays the number of users who use one, two, or three devices simultaneously to access the application. Users can be sorted by the maximum device cardinality.

Location Combinations When devices are used in combination, the URL locations that are viewed on each device are associated (Fig. 3). The locations are typically an indication of the content, state, or functionality of the application that a user is accessing.

Combined Views Ratio The combined views ratio is calculated by dividing the number of views of a location as part of device combination by the total number of views. This will result in a high number for locations that are mostly used in a cross-device setting and a low number for those that are mostly accessed from a single device.

Use Cases

The following use cases illustrate what kinds of questions an analyst may try to answer and how XD-Analytics supports them.

- Is there any cross-device usage of the application? How much time is spent online with cross-device and how much with single device usage? These questions can be answered by looking at device cardinalities (Fig. 2) which gives a first indication. Any cardinality higher than one denotes cross-device usage. A chart can be created that displays total time online for specific device cardinalities (Fig. 4). A similar chart can be created with average time online which could be used to determine if users with multiple devices spend more or less time with the application than single device users.

Figure 3. Location combinations, showing 47 users who accessed *home* and *landing* simultaneously.

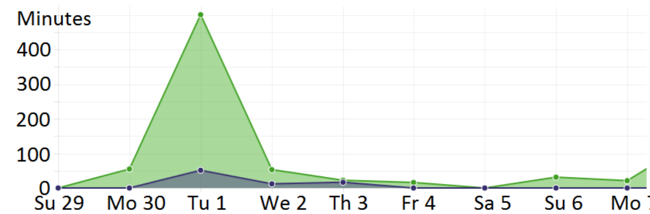

Figure 4. Time online for single device usage (green) and for combinations of two devices (dark grey)

- What devices are typically involved in parallel usage? Are there any unexpected combinations (e.g. three smartphones) that the application could be optimised for? On the other hand, are the expected combinations detected or are the users unaware that they could, for example, use the application with their mobile phone and their notebook? The device combinations, either as a bar chart (Fig. 1) or as an ordered list, provides answers to these questions.

- Which part of the application is commonly used with multiple devices? The analyst can investigate this by inspecting the most frequent location combinations. By filtering for a given device cardinality, they can discover which combinations are mostly accessed with, for example, two devices and which with three devices. Furthermore, the analyst can enter location patterns using regular expressions and filter for these patterns. This restricts the analysis to certain locations within the application. This is interesting, for example, if cross-device support is only available in certain locations.

- What devices in general are used to access the application? For an application without cross-device support, can we find indications where it could benefit from introducing cross-device functionality? The first question can be answered by looking at devices types, information on browsers, their versions, and operating systems used. The second question is harder to answer. However, inspecting device combinations and cardinalities as well as location combinations could provide some pointers. If an application has no specific cross-device support, but user still uses it with multiple devices simultaneously, this could be an indication that they would benefit from added cross-device support. Similarly, if there are more sessions than devices and thus users have multiple tabs open with the same application, it could be worth investigating if users would benefit from moving the additional tabs to a second device and coordinating interaction with the first one.

- What do users who combine a lot of devices do? What do users with specific combinations of devices (e.g. two tablets) do? These questions can be answered by filtering

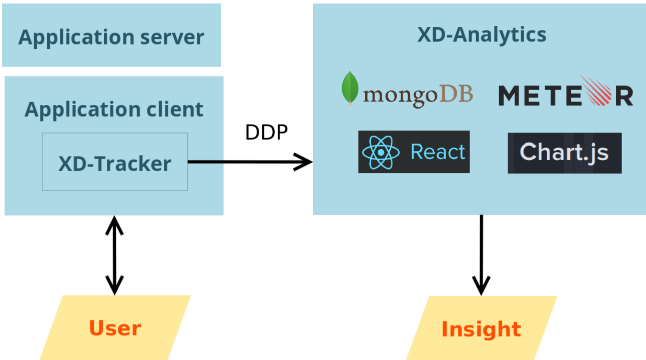

Figure 5. Architecture. A tracker integrated into the client application collects data and sends it to the analytics server.

the device combinations or cardinalities and then inspecting the locations. For example, the device cardinality could be set to the highest occurring number and then the locations and their combinations could be inspected. This would give an indication of what the user is doing with their devices. Likewise, a given combination (md, md) could be fixed and the process repeated.

ARCHITECTURE AND IMPLEMENTATION

XD-Analytics consists of a tracker that collects data and sends it to the analytics server where the data is processed and made accessible to the data analyst via a GUI. Figure 5 provides an overview of this architecture.

The tracker is a JavaScript file that needs to be embedded into the client application to be analysed. The tracker records events (such as location changes, user log ins and outs) as well as device characteristics (size, operating systems) and sends this information periodically to the server in a batch upload. Every 30 seconds, an artificial online event is generated to check if the user is still using the application and track the duration of a session. To associate devices with users, the client application should provide a user ID, which should not allow personal identification of the user to protect their privacy. The tracker persistently stores a unique ID for each device so that devices can be distinguished from each other and repeated visits with the same device can be recognised.

On the analytics server, the data is preprocessed. Event logs are aggregated into intervals of 10 minutes for performance reasons. These intervals are then aggregated by device and finally by user. From the aggregated data, the metrics introduced in the previous chapter can be computed. XD-Analytics is implemented based on the Meteor[6] platform and has been released as open-source software[7].

CASE STUDY

We used XD-Analytics to track the introduction of a cross-device feature into Taskbase[8], which is a learning platform that had no cross-device support when we started tracking. It

[6] https://www.meteor.com/
[7] https://github.com/mhusm/xd-analytics
[8] http://www.edtechlab.ch/taskbase

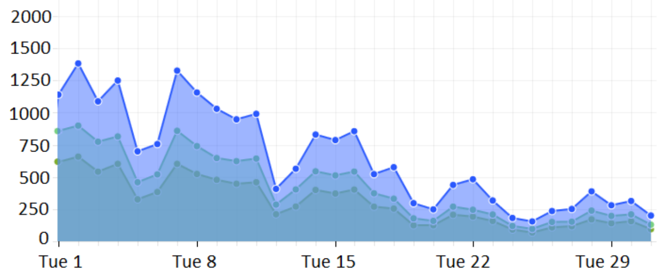

Figure 6. Sessions, users, and devices for Taskbase in December. Towards the end of the semester and on weekends usage is lower. There are more sessions than devices and more devices than users.

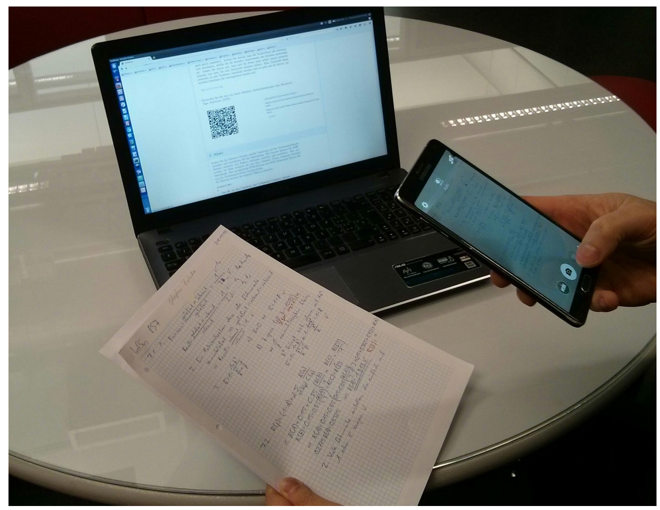

Figure 7. A Taskbase user first connects their phone by scanning a QR code on the notebook and then submits their solution by taking a picture.

has a user base of around 3500 users at ETH Zurich. Over the course of 9 weeks, we recorded more than 3 million log entries. In the beginning, we used XD-Analytics to get an idea of the users and the type and number of devices they use. Figure 6 provides an overview for the month of December 2015. The highest line denotes the number of sessions, the next one the devices, and the lowest the users. The figure shows that users access the application with more than one device on average and that they also use it in multiple tabs simultaneously. Further analysis showed that only a few small screen devices were used. This is likely due to the purpose of the application which is providing course material and exercise sheets to students. Medium or larger sized displays are more suited to such an application. A typical use case for Taskbase would be to display an exercise sheet with maths problems on a notebook, while writing the solutions by hand on paper. Later, the paper would then be handed in to an assistant for marking.

We addressed this last step by providing a cross-device feature for submitting solutions. A QR code is displayed on the exercise sheet which, when scanned with a mobile, logs the user into their account and lets them upload a solution for that exercise by taking a picture of their handwritten solution with the mobile (Fig. 7). As a pilot test, the feature was enabled in

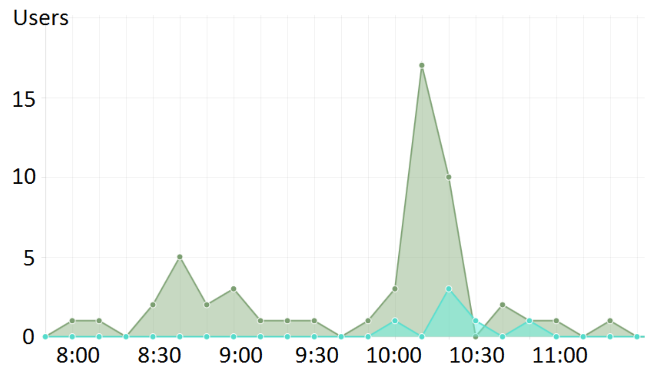

Figure 8. Device cardinalities for the morning of the feature introduction. The dark green line shows single device users, the light green one users with two devices.

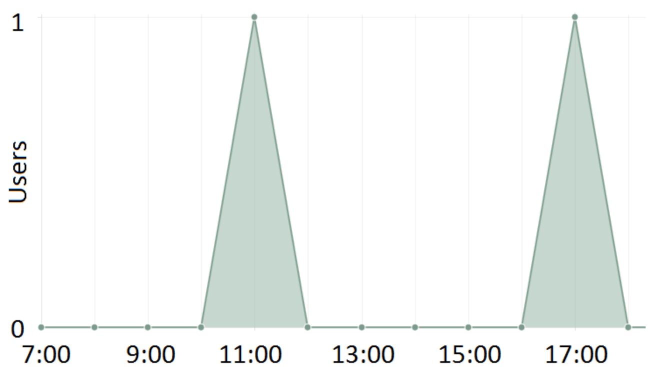

Figure 9. Cross-device usage for hand in. One student used the cross-device feature at 11.00, the other at 17.00 to hand in their solutions.

a single class that was usually attended by roughly 30 students. The feature was demonstrated to the students in an exercise session. Figure 8 shows the morning of that day with the data filtered using a location pattern to only show usage for locations associated with that class. The darker line denotes users with a single device, and the lighter one denotes those with two devices. The left half of the graph is for 8 to 10 am, when the students had a lecture for that class. A couple of students accessed the application with a single device during that time. At 10, the exercise session started and the cross-device feature was demonstrated. The figure shows that a couple of students tried it out, however, most kept using a single device.

The students had two weeks time to solve the problems. Figure 9 shows a day when two students handed in their exercises. The chart only shows how many users accessed the application with two devices, with single device usage not shown. The chart shows two instances of cross-device usage which matches the number of handed in solutions. Figure 11 shows the device combinations, again filtered for that specific class. It can be seen that the feature was mostly used as intended by combining a mobile phone (a small device) with a larger device such as a laptop or desktop computer.

When inspecting the combined views ratio (Fig. 10), we found the locations associated with the cross-device feature and the class where it was introduced at the top of the list. This demonstrates the value of this metric as it highlights locations

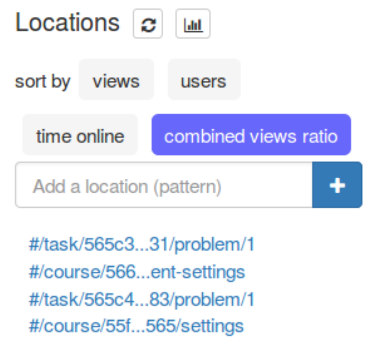

Figure 10. Locations sorted by the combined view ratio. Locations with high cross-device activity in relation to total activity are listed first.

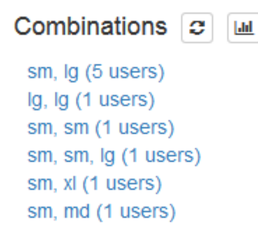

Figure 11. The device combinations observed in the class.

with high relative cross-device activity. The pilot class was rather small compared to other classes which could have 10 times the number of students. Consequently, other classes have a higher absolute number of accesses to locations and possibly also simultaneous use of multiple devices despite no specific support for this. The combined views ratio provides a relative measure and the result is as expected.

While inspecting Taskbase usage, we noticed a number of users who use two large screen devices simultaneously or have multiple tabs open. We assume that this is to display an exercise sheet and associated course material at the same time. This could be investigated further by looking at location combinations for this specific device combination. It could be an indication that users could benefit from the following cross-device support: When the user opens the exercise sheet, the application could automatically show the associated lecture on a second device or in a new tab.

Overall, XD-Analytics provided interesting insights into how Taskbase is used. Before the design and implementation of the cross-device feature, it provided us with a general overview of the type of devices that are used. Then XD-Analytics showed that the adoption of the cross-device feature was not very high, but that at least a few students used it. The system let us quantify the usage and track it over time. The metrics that we implemented in XD-Analytics allowed us to inspect various aspects of cross-device usage and the filters allowed us to focus only on the selected class in our analysis.

DISCUSSION

Our goal with XD-Analytics was to detected parallel multi-device usage patterns independent of any cross-device framework and based on the existence of a user concept in the application that is being tracked. The latter does not hold for all cross-device applications. In its current state, XD-Analytics cannot track applications that do not have a user concept and use an explicit pairing mechanism. Furthermore, if an application includes a user concept but applies a separate pairing mechanism, the explicit pairing will also not be tracked. That is, if the user is logged into the same application on two devices simultaneously, XD-Analytics will record it as parallel usage, whether the devices are paired or not. Similarly, cross-device usage between multiple users are currently not detected.

While these are limitations, detecting parallel usage based on user identity also has benefits. First, it is in line with the intuition behind what we understand by parallel usage: the same person using multiple devices simultaneously. Second, it does not rely on any cross-device framework and thus also supports the detection of parallel usage patterns in applications that have been built without any support for cross-device interactions. If such patterns are detected, they could be an indication of potential for cross-device interactions. However, further investigation in this direction is needed.

To address the explicit pairing of devices, XD-Analytics could be extended with a pairing event that developers would need to generate, similar to the existing log in event for users, and specify which devices have been paired. This would allow XD-Analytics to associate the paired devices and calculate the metrics that we introduced.

We can also see benefits in a tighter integration with cross-device frameworks to gain more detailed insights into how people are using cross-device applications. Some frameworks introduce user [16] or device [5] specific roles and these could be tracked as well. Others make use of the relative position of devices and/or users [15, 17]. Collecting and analysing such information could help us understand how the applications are used when deployed outside of lab environments.

Our focus has been on parallel patterns, however, the data that XD-Analytics records would be sufficient to also detect sequential patterns. Some sequential patterns can already be detected now by isolating a user and inspecting the devices they use with the timeline chart. However, this is quite cumbersome when done on a larger scale. To automate the finding of sequential patterns, the preprocessing step would need to be adapted so that, after grouping by user, the system would not only look for parallel but also for sequential patterns. Furthermore, the implementation currently has a granularity of 10 minutes and any activity falling into the same interval will be classified as parallel usage. While this is a limitation, our case study with Taskbase showed that performance needs to be taken into account for applications with a large user base and decreasing the interval would result in longer response times for queries.

XD-Analytics classifies devices according to their size as reported by the device. Devices could also be classified into categories such as phone, tablet, or notebook. However, the device landscape is now quite diverse. Large phones can easily reach the size of a small tablet and there are notebooks which double as tablets when the keyboard is removed. Rather than putting devices into these narrow categories, additional device characteristics could be recorded to provide more information on the type of device that is used. Such characteristics are sometimes used to determine how interfaces should be distributed across devices [24] and include input and output modalities such as touch pads, keyboards, and speakers.

Locations are used in XD-Analytics as indications of what functionality of an application a user is accessing or what state the application is in. Naturally, this is highly dependent on how an application makes use of URL locations. While using AJAX allows a web application to update its state without loading a new URL, it is considered good practice to update the history using the HTML5 History API[9] when the state changes. However, it is unclear how, or if at all, cross-device frameworks handle locations and history. As an alternative, a custom event for tracking state could be introduced, which would need to be emitted by the developer and thus require more effort than our current solution.

Finally, what constitutes device usage could be another point for discussion. A user could open an application on one device and then put that device aside without interacting. We attempt to identify such situations using heuristics based on user generated events (such as location changes). Even so, our system cannot distinguish a user who keeps the application open and does not pay attention to the device from one who, for example, spends a long time reading or watching a video without interacting. This is something that is addressed by ENGAGE [12] which can discern which of multiple active devices a user engages via face tracking. However, this comes at a cost in performance and privacy and the system has not been evaluated at scale. In comparison, XD-Analytics uses more lightweight tracking mechanisms while providing lower granularity for usage analysis.

CONCLUSION

We have investigated how usage of cross-device applications can be analysed in a lightweight manner. Inspired by techniques from web analytics, we have defined metrics of interest for cross-device applications and built a system that can track these, allowing the detection of parallel multi-device usage patterns. Compared to single device usage analysis, cross-device usage analysis adds complexity which we addressed with a flexible faceted user interface and filters. However, further effort could be put into designing cross-device specific visualisations. The system has been evaluated in a case study where we tracked in-the-wild usage of an educational platform with over 3000 users while a cross-device feature was introduced during the observed period.

ACKNOWLEDGEMENTS

This project was supported by grant No. 150189 of the Swiss National Science Foundation (SNF).

[9] https://developer.mozilla.org/en-US/docs/Web/API/History_API

REFERENCES

1. Sriram Karthik Badam and Niklas Elmqvist. 2014. PolyChrome: A Cross-Device Framework for Collaborative Web Visualization. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 109–118. DOI: http://dx.doi.org/10.1145/2669485.2669518
2. Lara D. Catledge and James E. Pitkow. 1995. Characterizing browsing strategies in the World-Wide web. *Computer Networks and {ISDN} Systems* 27, 6 (1995), 1065 – 1073. DOI: http://dx.doi.org/10.1016/0169-7552(95)00043-7
3. Karen Church, Denzil Ferreira, Nikola Banovic, and Kent Lyons. 2015. Understanding the Challenges of Mobile Phone Usage Data. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 504–514. DOI:http://dx.doi.org/10.1145/2785830.2785891
4. David Dearman and Jeffery S. Pierce. 2008. It's on My Other Computer!: Computing with Multiple Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 767–776. DOI: http://dx.doi.org/10.1145/1357054.1357177
5. Luca Frosini, Marco Manca, and Fabio Paternò. 2013. A Framework for the Development of Distributed Interactive Applications. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '13)*. ACM, New York, NY, USA, 249–254. DOI: http://dx.doi.org/10.1145/2494603.2480328
6. Linda Di Geronimo, Maria Husmann, and Moira C. Norrie. 2016. Surveying Personal Device Ecosystems with Cross-Device Applications in Mind. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays (PerDis '16)*. ACM, New York, NY, USA.
7. Giuseppe Ghiani, Fabio Paternò, and Carmen Santoro. 2012. Push and Pull of Web User Interfaces in Multi-device Environments. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, New York, NY, USA, 10–17. DOI:http://dx.doi.org/10.1145/2254556.2254563
8. Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. 2015. MultiFi: Multi Fidelity Interaction with Displays On and Around the Body. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3933–3942. DOI: http://dx.doi.org/10.1145/2702123.2702331
9. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-Device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2773–2782. http://doi.acm.org/10.1145/2556288.2557170
10. Maria Husmann, Michael Nebeling, Stefano Pongelli, and Moira C. Norrie. 2014. MultiMasher: Providing Architectural Support and Visual Tools for Multi-Device Mashups. In *Web Information Systems Engineering (WISE'14)*. Springer International Publishing, 199–214. http://dx.doi.org/10.1007/978-3-319-11746-1_15
11. Tero Jokela, Jarno Ojala, and Thomas Olsson. 2015. A Diary Study on Combining Multiple Information Devices in Everyday Activities and Tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3903–3912. DOI: http://dx.doi.org/10.1145/2702123.2702211
12. Rachel Jones, Sarah Clinch, Jason Alexander, Nigel Davies, and Mateusz Mikusz. 2015. ENGAGE: Early Insights in Measuring Multi-Device Engagements. In *Proceedings of the 4th International Symposium on Pervasive Displays (PerDis '15)*. ACM, New York, NY, USA, 31–37. DOI: http://dx.doi.org/10.1145/2757710.2757720
13. Fahim Kawsar and A.J. Bernheim Brush. 2013. Home Computing Unplugged: Why, Where and when People Use Different Connected Devices at Home. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 627–636. DOI:http://dx.doi.org/10.1145/2493432.2493494
14. Dejan Kovachev, Dominik Renzel, Petru Nicolaescu, and Ralf Klamma. 2013. DireWolf - Distributing and Migrating User Interfaces for Widget-Based Web Applications. In *Web Engineering (ICWE'13)*. Springer Berlin Heidelberg, 99–113. http://dx.doi.org/10.1007/978-3-642-39200-9_10
15. Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. 2011. The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 315–326. http://doi.acm.org/10.1145/2047196.2047238
16. Michael Nebeling, Theano Mintsi, Maria Husmann, and Moira Norrie. 2014. Interactive Development of Cross-device User Interfaces. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2793–2802. DOI: http://dx.doi.org/10.1145/2556288.2556980
17. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 45–54. DOI: http://dx.doi.org/10.1145/2669485.2669500

18. Stephanie Santosa and Daniel Wigdor. 2013. A Field Study of Multi-Device Workflows in Distributed Workspaces. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 63–72. http://doi.acm.org/10.1145/2493432.2493476

19. Dominik Schmidt, Julian Seifert, Enrico Rukzio, and Hans Gellersen. 2012. A Cross-device Interaction Style for Mobiles and Surfaces. In *Proceedings of the Designing Interactive Systems Conference (DIS '12)*. ACM, New York, NY, USA, 318–327. DOI: http://dx.doi.org/10.1145/2317956.2318005

20. Mario Schreiner, Roman Rädle, Hans-Christian Jetter, and Harald Reiterer. 2015. Connichiwa: A Framework for Cross-Device Web Applications. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 2163–2168. DOI: http://dx.doi.org/10.1145/2702613.2732909

21. Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explor. Newsl.* 1, 2 (Jan. 2000), 12–23. DOI: http://dx.doi.org/10.1145/846183.846188

22. Lucia Terrenghi, Aaron Quigley, and Alan Dix. 2009. A Taxonomy for and Analysis of Multi-person-display Ecosystems. *Personal Ubiquitous Comput.* 13, 8 (Nov. 2009), 583–598. DOI: http://dx.doi.org/10.1007/s00779-009-0244-5

23. Daniel T. Wagner, Andrew Rice, and Alastair R. Beresford. 2014. Device Analyzer: Large-scale Mobile Data Collection. *SIGMETRICS Perform. Eval. Rev.* 41, 4 (April 2014), 53–56. DOI: http://dx.doi.org/10.1145/2627534.2627553

24. Jishuo Yang and Daniel Wigdor. 2014. Panelrama: Enabling Easy Specification of Cross-Device Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2783–2792. http://doi.acm.org/10.1145/2556288.2557199