Assisted Pattern Mining for Discovering Interactive Behaviours on the Web

DOI:
10.1016/j.ijhcs.2019.06.012

Document Version
Accepted author manuscript

Assisted Pattern Mining for Discovering Interactive Behaviours on the Web

Aitor Apaolaza, Markel Vigo*

School of Computer Science, University of Manchester

Abstract

When the hypotheses about users' behaviour on interactive systems are unknown or weak, mining user interaction logs in a data-driven fashion can provide valuable insights. Yet, this process is full of challenges that prevent broader adoption of data-driven methods. We address these pitfalls by assisting user researchers in customising event sets, filtering the noisy outputs of the algorithms and providing tools for analysing such outputs in an exploratory fashion. This tooling facilitates the agile testing and refinement of the formulated hypotheses of use. A user study with twenty participants indicates that compared to the baseline approach, assisted pattern mining is perceived to be more useful and produces more actionable insights, despite being more difficult to learn.

Keywords: Interaction logs, Assisted pattern mining, User interface evaluation

1. Introduction

Understanding users' interaction with complex interactive systems is a challenging endeavour. While task-oriented user evaluations help to optimise the user interface elements involved in the execution of known tasks, user behaviour beyond the established boundaries of the tasks remains unknown. This pragmatism is understandable in that evaluating all possible tasks is not feasible. Alternatively, data-driven approaches enable data-savvy specialists to identify the

*Corresponding author

Email addresses: aitor.apaolaza@manchester.ac.uk (Aitor Apaolaza), markel.vigo@manchester.ac.uk (Markel Vigo)

emerging patterns of use on logs containing user interaction data. For instance, given a dataset of interaction events, sequential pattern mining algorithms find the most frequent sequences of events (Mooney and Roddick, 2013). Following similar approaches, several works explore the extraction of event sets from user interaction logs for isolating the regularities exhibited by users (Dev and Liu, 2017; Perer and Wang, 2014; Sarkar et al., 2016; Zgraggen et al., 2015). While fine-grained user interaction log data provides extensive details about users' interaction, mining such data is a complex task, posing various challenges:

Challenge 1: High-cardinality. The high number of unique user interaction events makes the selection of event sets from raw data an overwhelming task. Grouping techniques have been proposed to reduce high frequency events such as mouse movements and scroll (Chudá et al., 2018). Subsetting and transforming the input is pertinent when there are particular events that might not be relevant for the evaluation of the task or fall outside the scope of the user interface to be evaluated (Dev and Liu, 2017). For example, if the objective was to compare different areas of interest on a website, subsetting would enable to separately evaluate the interactions on these areas and their surrounding interactions (**?**).

Challenge 2: Limited semantics. Raw user interaction events lack a rich context of use from which one can extract meaningful conclusions. To increase this lack of meaning, events should be associated with elements on the website and mapped into the appropriate abstraction levels (Hilbert and Redmiles, 2000; Liu et al., 2017; Perer and Wang, 2014). This would allow, for instance, to transform mouse clicks on a specific element of a Web page (i.e. *mouse click on a button*) into semantically richer events (e.g. *submit search query*).

Challenge 3: Noisy outputs. Pattern mining algorithms generate a large number of resulting patterns that require being filtered to facilitate decision making (Seno and Karypis, 2002). The discovery of useful patterns is non-trivial, and domain knowledge is necessary to associate the output of the pattern mining algorithms with actual tasks and behaviours (Dev and Liu, 2017). The abstraction level of the events used as input should be tailored to the purpose of the evaluation as key details that help to interpret the results may be missed

otherwise. For example, while the analysis of mouse movement events might be useful to discern how users' allocate their attention on the screen, their high frequency would minimise the prominence of less frequent events such as mouse clicks.

Challenge 4: Identifying complex and outlying behaviours. Pattern mining techniques favour reoccurring scenarios. Consequently, the results follow a majority rule, where the most common patterns are the candidates for further exploration. However, the purpose of the evaluation might be focused on less frequent (but still relevant) activities. Unexpected interaction patterns may indicate usability problems, and unusual and unforeseen uses of the user interface (Akers et al., 2009). Unfortunately, current approaches lack support for identifying and understanding outlying behaviours.

Challenges 1–3 are related in that the granularity and semantics of the event sets used as input for pattern mining algorithms (*Challenge 1 and 2*) determines the interpretability of the resulting patterns, i.e. *Challenge 3* (Hilbert and Redmiles, 2000). In order to handle noisy outputs while increasing meaning, one strategy can be to reduce the number of input events and enrich their semantics, which needs human intervention to tune the entry parameters and find the right abstraction level.

1.1. Workflows for Interactive Log Mining

According to Pirolli and Card (2005), extracting knowledge from raw user interaction data calls for agile analysis driven by data (i.e. bottom-up processes) or theory (i.e top-down processes). These two non-exclusive approaches are affected by the above-mentioned challenges when choosing the right granularity and semantics of the data, reformulating current hypotheses based on the outcomes of earlier evaluations, and refining the analysis so that meaningful interaction patterns can emerge. Informed by Fayyad et al. (1996), we introduce data wrangling functionalities and software infrastructure that enable such iterative analysis, while addressing the above-mentioned challenges:

- **Subsetting** user interaction event data to select the event sets to be used for pattern mining.
- **Scoping** the context of use of the selected events, where context is defined as the element(s) on a Web page that trigger such event (e.g. mouse hover on a picture) as well as the specific URLs of the Web pages.
- **Mapping** tools to combine low-level events and transform them into semantically more meaningful actions.
- **Defining hypotheses of use** that specify complex user behaviours through sequences of events. User researchers can define a set of custom events and set time constraints between them in order to retrieve such behaviours from the raw data: e.g. a mouse hover on a picture that lasts more than 5 seconds after scrolling more that one third of the screen.
- **Refining** the event set used as input for pattern mining algorithms informed by earlier outputs. Since this strategy entails to gradually add/remove events, we provide an efficient hypothesis testing engine that enables quick turnarounds.

While the mentioned challenges can be addressed (not without difficulty) by specialists who master the use of pattern mining techniques, they certainly represent a barrier to individuals who are knowledgeable about human factors on the Web but are discouraged by the complexities of data wrangling and pattern mining (i.e. user researchers). In this paper we reduce these barriers using tools to facilitate the adoption of pattern mining techniques by a wider range of individuals. To that end, we introduce two tool-supported workflows that use the above functionalities to support the discovery of interactive behaviours on the Web. Using the framework defined by Pirolli and Card (2005) our workflows implement bottom-up functionalities in order to derive sequences of interest from the data, and enable introducing hypotheses in a top-down fashion. The **assisted** workflow allows user researchers to guide the execution of pattern mining algorithms by customising the event set to be used as input of the algorithms and iteratively add/remove custom events to refine the results, choosing the appropriate granularity of the events as they reformulate their hypotheses. The

assisted++ workflow extends the assisted workflow by supporting the definition and testing of custom hypotheses on the event set. Data analysts have been found to perform similar workflows to the above, where they iteratively query and mine event sequences to understand user behaviour (Law et al., 2018).

These two analysis workflows support user researchers in formulating hypotheses that might be considered weak or could even be mere *expectations*. Nevertheless, these hypotheses serve as a starting point that inform the initial exploration of data. Then researchers can iterate from expectations to consolidated hypotheses, which can be tested in experiments and A/B tests. The contributions of this paper are two-fold:

- We extend WevQuery (Apaolaza and Vigo, 2017) with a set of functionalities that address the challenges of mining low-level user interaction event logs. We call this new enhanced version WevQuery for Pattern Mining, i.e. WevQuery-PM.
- We evaluate the trade-off between the added complexity of these functionalities and their usefulness. The results of a user study with twenty participants suggest that even though the proposed workflows were more difficult to learn than a baseline workflow without tool support, they enabled user researchers to come up with more useful and actionable insights.

2. Related Work

Web server logs typically include clickstreams, which enable the analysis of Web traffic and timings (Srivastava et al., 2000). Beyond clickstreams, fine grain user interaction logs can tackle problems inherent to Web server logs, such as automatic page reloads incorrectly interpreted as user interaction (Weinreich et al., 2006). Yet, the richer the data is, the more complex it is to analyse, requiring individuals with data wrangling skills, or tools that process and visualise

data. Popular tools such as Google Analytics[1], Woopra[2] or Matomo[3] capture clickstreams from users and provide aggregated data of demographics, landing pages and most frequent transitions between pages of a website. Visual analysis of these clickstreams helps to identify large volumes of traffic and compare user behaviours over time (Carta et al., 2011; Zhao et al., 2015). Relevant user interaction events can be included into the aggregated visualisations to provide a more detailed view of the users' path through the website. For example, the mentioned tools visualise particular events such as "Start chat" or "Signup".

Common tasks and user flows can be extracted from user interaction sequences (Deka et al., 2016). The ideal path to be taken can be defined as the *golden trace*, enabling the isolation of interactions that deviate from this path (Deka et al., 2017). Then users with similar clickstream patterns can be grouped into stereotypical personas who use the system (Zhang et al., 2016). The recreation of particular Web interaction recordings of individual users allows developers to find hard to replicate behaviours (Burg et al., 2013). When the task being performed is known, visualisations of finer grained interaction, such as mouse clicks, enable comparisons between various user sessions (Rzeszotarski and Kittur, 2012; Breslav et al., 2014; Paternó et al., 2016).

2.1. Pattern Mining on User Interaction Logs

Pattern mining is typically employed to discover regularities in a data-driven fashion. The use of pattern mining to extract frequent itemsets was initially found useful to isolate such regularities in shopping behaviours (Borgelt, 2012), where frequent itemsets would refer to a set of items that are frequently purchased together. In the case of user interaction log analytics, frequent itemsets can refer to events taking place in the same session or a single visit to a website. Specifically, sequential pattern mining algorithms (Mooney and Roddick, 2013) compute the frequently occurring subsequences in a dataset of sequences,

[1] `https://analytics.google.com`
[2] `https://www.woopra.com/`
[3] `https://matomo.org/` (previously known as Piwik)

whereby the *support* parameter indicates an occurrence threshold above which, the discovered patterns are reported (Mannila et al., 1997). Pattern mining algorithms stop their execution when all the patterns above the given support threshold are found.

The relevance of the patterns produced by pattern mining algorithms is dependent on the domain and context of use and, therefore, reliant on the event set used as input (see *Challenge 1: cardinality*), as well as subject to experts' opinion about their usefulness. A pattern could be considered *useful* if it is unknown for the researcher and the finding is actionable, i.e. they can use it to their advantage (Silberschatz and Tuzhilin, 1995). The output of pattern mining algorithms is typically large (see *Challenge 3: noise*), and pruning and ranking such output is necessary to help find relevant patterns. The length of a pattern can be used in combination with its support as a criterion to judge its relevance. However, it can be argued that short patterns with high support can be as relevant as longer patterns with smaller support (Seno and Karypis, 2002). Alternatively, techniques such as *membership based cohesion* (Dev and Liu, 2017) rank the sequences by comparing the frequency of the events in a given pattern in other candidate patterns. User-defined filters have also been employed: for example, in the case of sequences of timestamped locations, only the patterns involving stays in a particular place for a given amount of time were sought (Law et al., 2018). The analysis of event sequences has been found useful to acquire insights into users' interaction (see *Challenge 4: complexity*). Under certain conditions, specific sequences of user interaction events are indicators of problematic behaviours (Vigo and Harper, 2017) and usability problems (de Santana and Baranauskas, 2015). For example, successive interaction repetitions (Li et al., 2010) and the use of corrective functionalities such as *undo* (Akers et al., 2009) can be used to detect possible usability problems.

Human intervention is often needed to determine the relevance of machine-generated results, such as classifying extracted patterns into typical tasks (Dev and Liu, 2017). In the case of high-volume data, reducing waiting times during computations is extremely critical in order to support human-driven iterative

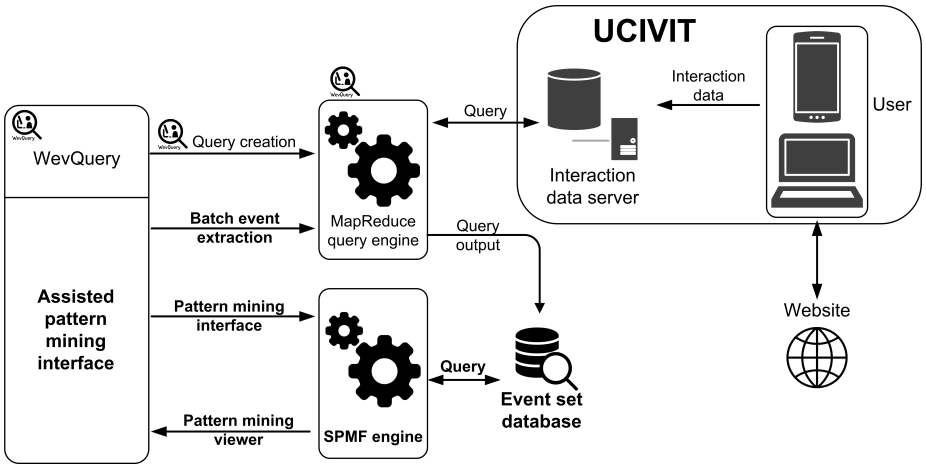

Figure 1: Architecture of the assisted pattern mining tool

analyses (Malik et al., 2016). The use of interactive visualisations can help addressing the problem of noisy outputs. *Frequence* supports the exploration of varying levels of detail of the resulting patterns by combining pattern mining algorithms with the use of increasingly detailed dictionaries (Perer and Wang, 2014). The results that align with the goals of the analysis are isolated, while less relevant results are filtered out. In particular, enabling the selection of relevant events helps to tailor the presentation of the results so that relevant transitions between the events can be highlighted in the resulting visualisations (Liu et al., 2017).

3. Assisted Pattern Mining: Architecture and Workflows

We describe the architecture of the system that implements the workflows for mining Web interaction logs. Our proposal in Figure 1 reuses two components for data logging and querying.

Logging user interaction data. We use UCIVIT (Apaolaza et al., 2013) to capture Web interaction events and store them in a remote NoSQL database[4] (i.e.

[4]GitHub repository: `github.com/aapaolaza/UCIVIT-WebIntCap`

the "Interaction data server" in Figure 1). Table 1 shows a sample of the events and the type of contextual information retrieved. For example, `mousedown` events contain the information about the user interface element the user clicked on. In the case of mobile events, the coordinates for all the available inputs from the multitouch interface are also captured. For each event, UCIVIT extracts information such as the `ID`, `type`, `class`, and text content. The URL is also captured with and without GET parameters, so interaction within similar URLs can be grouped together (e.g. search results page).

Querying user interaction data. We use WevQuery (Apaolaza and Vigo, 2017) to test hypotheses about users' interaction by defining queries as a sequence of single or multiple events (we give further details in Section 3.1). These queries are transformed into scalable MapReduce queries to be run against the "Interaction data server", extracting all the occurrences of the described sequence of events[5].

These two components are loosely coupled: WevQuery could work with any user interaction dataset provided that data is timestamped. The output of WevQuery queries gets stored in the "Event set database", which becomes the input for pattern mining algorithms. The "Assisted pattern mining interface" module extends WevQuery, to implement the tool supported workflows with the following functionalities:

- The "Batch event extraction" functionality, which is described in further detail in Section 3.2, automatically generates a set of queries to extract customised inputs for pattern mining (e.g. all occurrences of individual `mousedowns` and `mouseups` on a set of interface targets). Since user interface targets can be identified via their `ID`, `class` or `type`, the `ID` selector is given priority, so type and class selectors will only match elements without any associated IDs. In the case of `type`, aliases are used to make these

[5]The original WevQuery system is marked with the WevQuery icon (🔍) in Figure 1.

targets easier to understand by non-experts (e.g. *image* instead of `img`). After setting the queries up, they are automatically processed in the same way as the WevQuery hypotheses, and their results would be available in the "Event set database".

- The "Pattern mining interface" allows users to select the input and tune the parameters of the pattern mining algorithms (see Section 3.3). The "SPMF engine" queries the events sets (i.e. results of queries) from the "Event set database", merges and sorts them based on their timestamps, and transforms the resulting sequences to be amenable to SPMF, the data mining library. Pattern mining algorithms are launched on these event sets and the results are then visualised through the "Pattern mining viewer".

3.1. Interactive Hypothesis Formulation and Testing

WevQuery provides an interactive Web application to support user researchers in the creation of hypotheses about Web interaction. Figure 2 shows the "Query creation" view displaying the *Event Palette* (A), containing the user-generated *Event Matching Blocks* (B), which define *custom events*: each matching block contains an event (e.g. `mouseover`) and an optional context for this event, which indicates the element (or the set of elements) of the user interface that triggers such event. The link between the trigger and the subsequent event is explicitly established by associating the event with the particular properties of the element including label-value pairs such as the ID and class HTML attributes. This functionality supports the creation of *custom events* as a combination of a low level event and its context addressing at the same time *Challenge 1: cardinality* and *Challenge 2: semantics*.

These blocks defining *custom events* can be dragged from the *Event Palette* (A) to the *Hypothesis Formulation* (C) container in order to specify the sequence of event blocks to be extracted. The order of the events as well as temporal binary relationships can be set between event blocks to define complex sequences of events that have time constraints in (D). Since these complex sequences can be

Type	Events	Description	Additional information	Target
Mouse	`mousedown`	Start of mouse click action	Coordinates	**X**
	`mouseup`	End of mouse click action	Coordinates	**X**
	`mousemove`	Mouse movement	Coordinates	**X**
	`mouseover`	Hovering into target	Coordinates	**X**
	`mouseout`	Hovering out from target	Coordinates	**X**
	`doubleclick`	Double mouse click	Coordinates	**X**
	`mousewheel`	Mouse wheel interaction	Scroll distance	**X**
Selection	`select`	Selection of page content	Text content	**X**
	`cut`	Content cut	Text content	**X**
	`copy`	Content copy	Text content	**X**
	`paste`	Content paste	Text content	**X**
Keyboard	`keydown`	Start of key press action	Pressed key	**X**
	`keyup`	End of key press action	Pressed key	**X**
	`keypress`	Key press action	Pressed key	**X**
Window	`load`	Page is loaded	Window size	
	`resize`	Browser window is resized	Window size	
	`unload`	Window is closed		
	`windowfocus`	Browser tab gains focus		
	`windowblur`	Browser tab loses focus		
	`scroll`	Change of scroll state	Scroll distance	
Mobile	`touchstart`	Start of touch screen action	Multitouch coordinates	**X**
	`touchend`	End of touch screen action	Multitouch coordinates	**X**
Other	`change`	Input element state change	New value	**X**
	`contextmenu`	Opening of context menu	Coordinates	**X**

Table 1: Sample of the events captured by UCIVIT

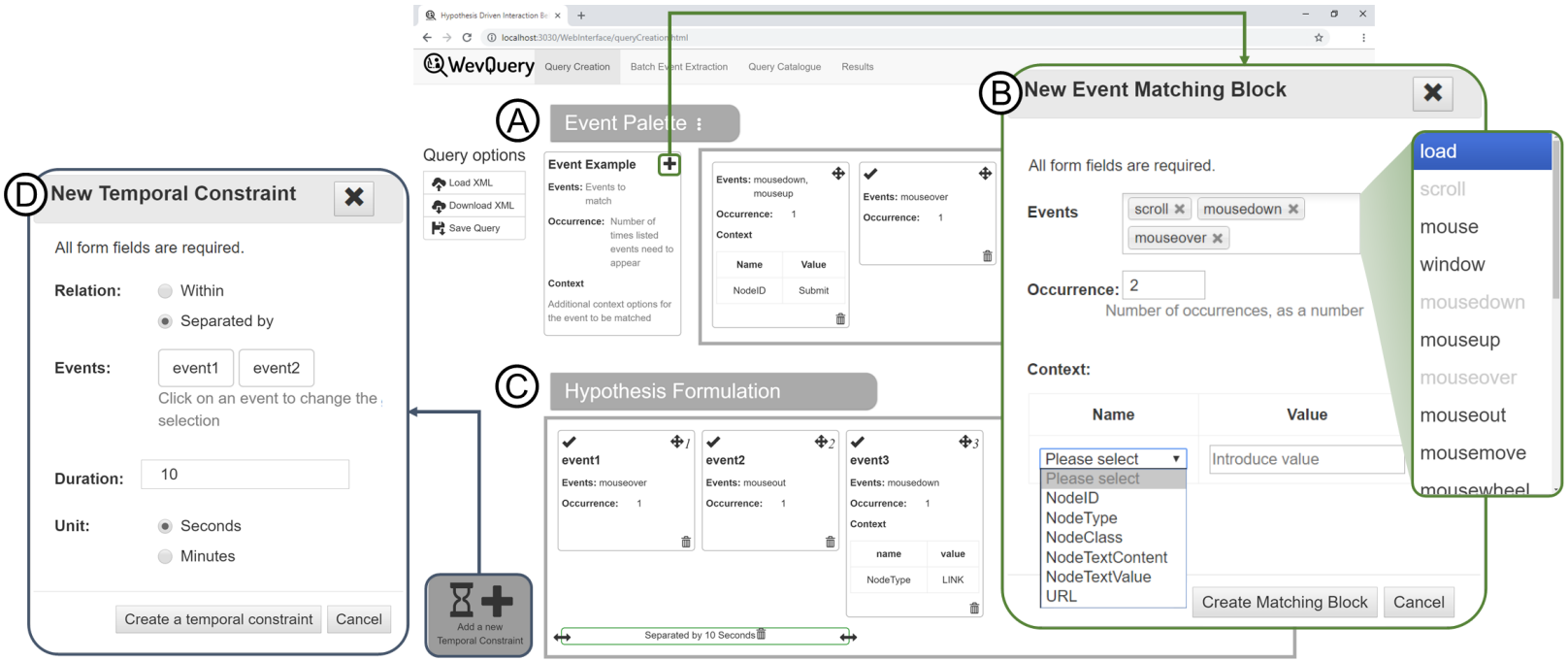

Figure 2: The "Query creation" view

conceived as micro-behaviours (or micro-interactions as defined by Breslav et al. (2014)) that are exhibited on the Web, this functionality addresses *Challenge 4: complexity* in that it supports the formulation of hypotheses of interactive behaviour. For instance, Figure 2 illustrates how to build an hypothesis to seek all the instances of users hovering any interface element for longer than 10 seconds, followed by a click on an element of type `LINK`. This behaviour could help identify instances of users struggling to find a specific link, or interacting with a hover-activated element to disclose more information. The sequence defining a hypothesis of Web use can then be run as a query against the database storing the users' interaction data, yielding all the occurrences of a given hypothesis.

In order to enable quick turnarounds when testing the hypotheses on large datasets, the system implements the MapReduce programming paradigm (Dean and Ghemawat, 2008). Once a query is created, it can be saved and accessed through the *Query Catalogue* ① menu depicted in Figure 3. Then, user researchers can select queries from this list and run them against the database of interaction data. The results of the queries are stored in the "Event set database", which contains all the occurrences of the formulated hypotheses alongside the context in which the occurrence took place: the URL, the user identifier and detailed information of the events involved. On a live website, if the ID elements remain unchanged, the same query can be run on a dataset that is being constantly updated. Users can then explore the results of completed queries on the *Query Results* ② component in Figure 3 to visualise particular results, or to include them as input for pattern mining algorithms.

3.2. Batch Event Extraction

While researchers can formulate complex hypotheses in the "Query creation" view, they do not always have specific working hypotheses. The "Batch event extraction" view in Figure 4 helps researchers who do not have absolute certainty about how a website is being used in narrowing down their search through a number of functionalities that enable subsetting events, setting their scope and mapping them into more meaningful events:

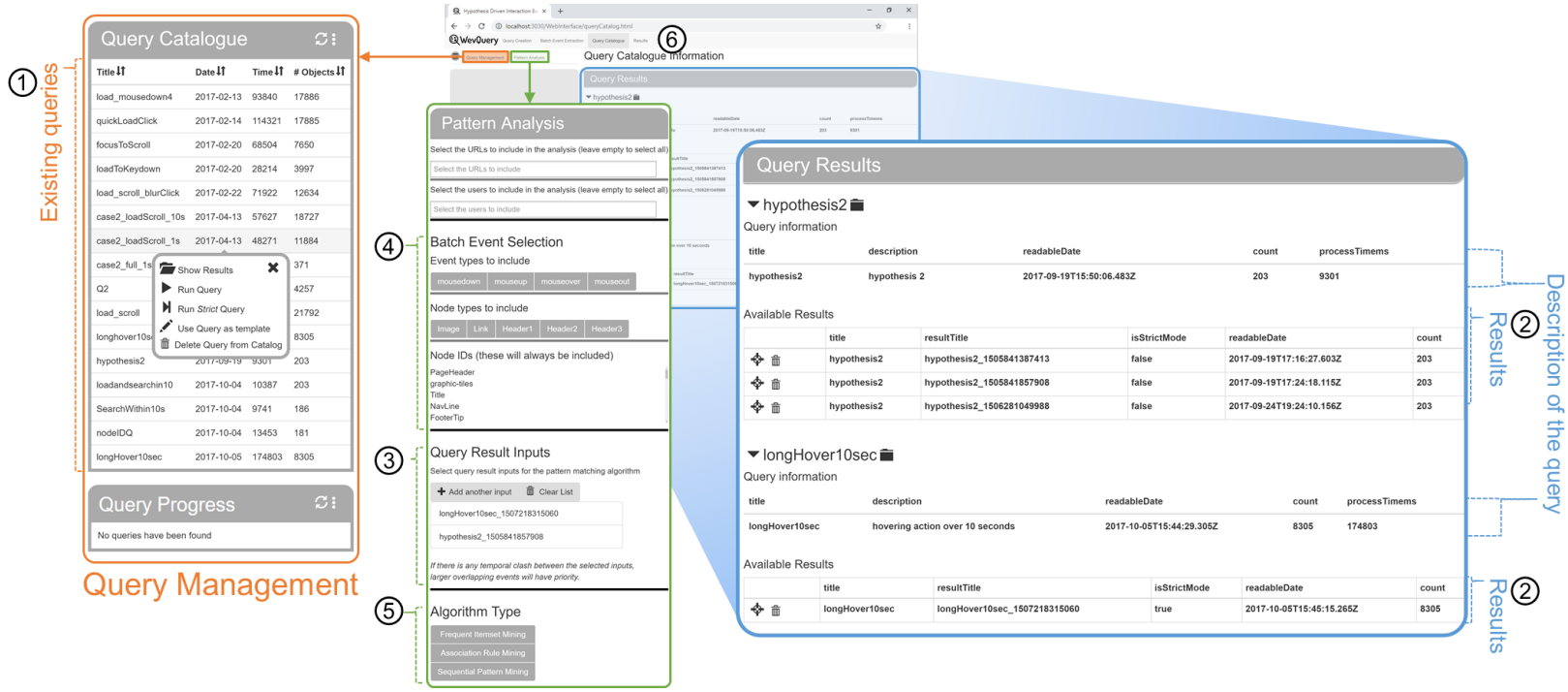

Figure 3: The "Pattern Mining" view including the *Query Catalogue*, *Pattern Analysis* and the *Query Results* components

1. In order to facilitate the selection of subsets of events, WevQuery-PM extracts automatically all the unique events that exist in the dataset. User researchers can select the events of interest with a mouse click on the text field labelled as 'Event list'.

2. Similar to the above step, all the unique ID attributes are automatically extracted so that user researchers can define the scope of the events. The text field 'ID list' in Figure 4 enables the selection of ID attributes of interest.

3. In addition to setting the scope of the events using their ID attribute, the scope of events can be further defined by selecting the HTML elements of interest. By default, in order to increase readability, we provide aliases so that the user can select images, hyperlinks and headers, which correspond to the `IMG`, `A` and `H1`–`H3` HTML elements respectively. Users can create further aliases by defining custom events.

3.3. Incorporating Pattern Mining into the Analysis Workflows

The "Pattern Analysis" component in Figure 3 allows users to select the event sets that are going to be used as input for the pattern mining algorithms and address *Challenge 1: cardinality* and *Challenge 3: noise*. As we have seen so far, there are two ways to create event sets. Users can use *Query Inputs* ③ elements to select as input the results of any of the previously executed queries from the *Query Results* ② component. These queries can be either *custom events* or more complex hypotheses, and might have been created by other users of the system. Alternatively, the system automatically extracts a set of events from the raw interaction data through the "Batch Event Selection" functionality. A view of the latter is available under the corresponding header ④ in the *Pattern Analysis* component which includes the events and user interface elements discussed earlier. Once the corresponding inputs (both *Query Inputs* ③ and *Batch Event Selection* ④) are selected, users can choose which pattern mining algorithms to run ⑤. Then the user can set the parameters of these

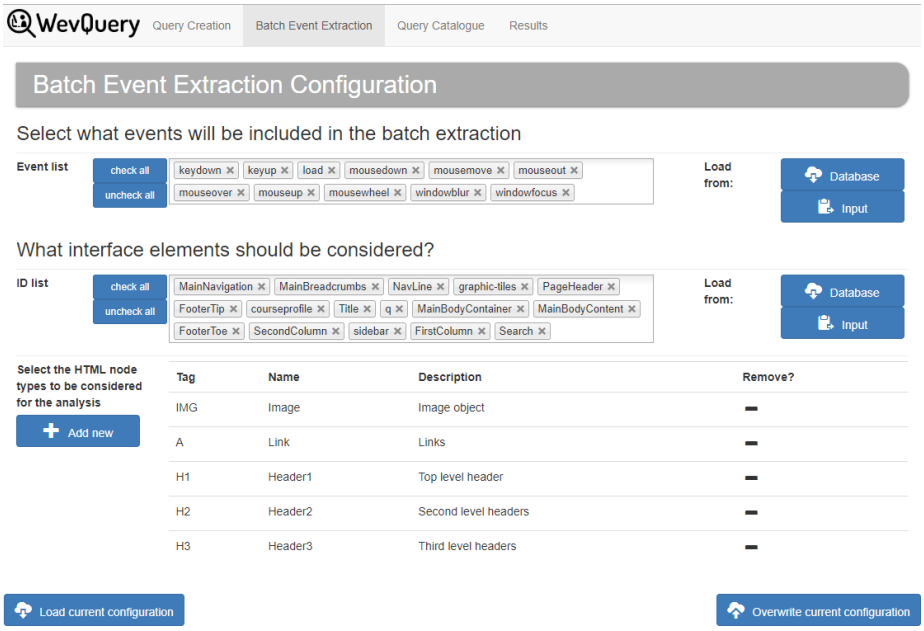

Figure 4: The "Batch event extraction" view

algorithms, such as their minimum support and minimum confidence thresholds. When users launch the analysis, all the selected inputs are retrieved from the database and put together into an event set that is pipelined into the pattern mining algorithms.

WevQuery-PM integrates the SPMF library (Fournier-Viger et al., 2016), an open source data mining library. SPMF takes a formatted text file as input, and prints the outcome of the selected pattern mining algorithm into another text file. In WevQuery-PM, we have included the Apriori algorithm (Agrawal et al., 1994) for frequent itemset mining and the PrefixSpan algorithm (Han et al., 2001) for sequential pattern mining. When the execution of the algorithm is completed, a new tab ⑥ opens up next to the top tabs (indicated in Figures 3 and 5), where SPMF's output is channelled to display the patterns found, ranked in descending order according to their frequency.

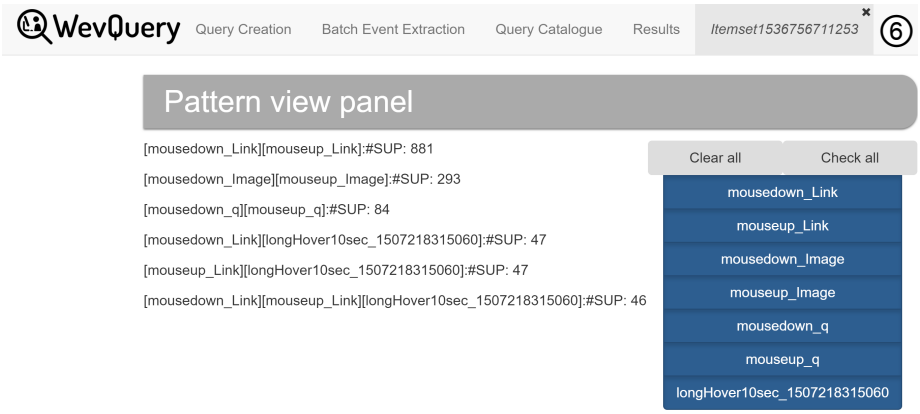

Figure 5: Output of the frequent itemset algorithm applied to an event set formed by all available mouse clicks (`mousedown` and `mouseup`) and the `longHover10sec` hypothesis

3.4. Supported Analysis Workflows

The output of the pattern mining algorithms generate patterns that enable users to explore how the formulated hypotheses relate to the selected inputs. Any discoveries made by the user can then be used to inform the definition of a new event set, supporting an iterative analysis of the data. This way, WevQuery-PM supports two different analysis workflows, which we name assisted and assisted++, according to the support user researchers are given and their complexity.

In the assisted workflow, users can increase the chances of finding meaningful patterns by customising the event set to be used as input for pattern mining. The *Batch Event Selection* can be used to modify the input, focusing on particular combinations of events. *Custom events* created using the "Query creation" view by any user of the system, can also be included in the event set. The scope of the analysis can also be defined choosing which URLs (or subset of URLs) are to be included in the pattern mining analysis. For example, the user can test if the occurrence of a particular pattern is limited to a particular Web page, or extend the analysis to all URLs of the website.

In addition to customising the event sets, the assisted++ workflow enables

users to include hypotheses in the analysis workflow. The use of already existing interaction hypotheses, represented as queries, helps to determine not only if a particular hypothesised interaction occurs, but also to explore the context in which it happens.Results corresponding to various queries can be included in the analysis, allowing users to explore relations between hypotheses. Additional hypotheses can be designed and included in the workflow using the "Query creation" view (see Figure 2).

4. Evaluation

Following the snowball sampling technique to recruit participants, twenty individuals (10 female, 10 male, median age 29.5, SD=4.82, fifteen computer scientist, two psychologists, one business school student, one social scientist, and one telecommunications engineer) took part in a user study to evaluate the trade off between the complexity of assisted pattern mining workflows and the knowledge acquired through their use in WevQuery-PM. While we are aware that it is risky to generalise about professional skills, we sought individuals with a skill set similar to that of user researchers or designers.

Participants reported their confidence about various topics on a range from 1 (unconfident) to 4 (confident). Participants' confidence of UX (median = 3, SD = 0.72, ▟) and Web markup languages (median = 3, SD = 0.88, ▟) was high, while their confidence of pattern mining techniques (median = 2, SD = 1.14, ▙) was lower. Our sample represented individuals who were experienced in Web technologies and knowledgeable about human factors on the Web but lacked the skills to use pattern mining tools to conduct sophisticated analyses on the data. Hence, participants played the role of a user researcher who was willing to use pattern mining algorithms to get further insights into the usage of the user interface on a large website, but lacked the necessary data processing and pattern mining skills. The participants performed the tasks in Table 2 using the workflows described in the previous section. Following a think-aloud procedure (Lewis, 1982), the first author took notes of the feedback given by

Table 2: Tasks given to the participants in the study

Workflow	**Task**	**Prompt**
Non-assisted	Guided	Work with a predefined set of events. Simulate the situation where an expert in data processing has extracted a series of sequences of events for you to analyse.
	Exploration	Based on the results of the guided task above, try interpreting the results shown to you.
	Directed	Run the Frequent Itemset Mining algorithm again, with the same input, but this time limit the minimum Support threshold to 40%.
Assisted	Guided	Generate your own dataset to use as input for pattern mining. Choose which events will be included (`mousedown`, `mouseup`, `mouseover`, and `mouseout`), as well as the user interface element upon which such events were triggered.
	Exploration	Based on the results of the guided task above, try interpreting the results shown to you.
	Directed	Run the Frequent Itemset Mining algorithm, but this time try to answer the following question. How many times do users click on an image AND a link during the same episode?
Assisted++	Guided	Test Hypothesis 1: Users hover over the same element for longer than 10 seconds, either because they are triggering an interactive event (such as disclosing dropdown dialogues) or as part of exploring the interface.
	Exploration	Based on the results of the guided task above, try interpreting the results shown to you.
	Directed	Test Hypothesis 2: Within 10 seconds from loading the page, users start the action of clicking on the search bar.

participants as well as any insight they reported while carrying out the tasks. Two months of interaction data on website of the School of Computer Science, University of Manchester[6] were used as stimuli of the study, accounting for a total of 5.7m low-level events generated by 2 445 unique users, who generated 9 862 interaction episodes. We define an episode as a continuous interaction without interruptions that are longer than 40 minutes, in line with what the literature suggests about the length of sessions (Heer and Chi, 2002). This website followed modern Web standards, as was the home page of a school from a university that attracts thousands of visitors every month. A colour printout of the screenshot shown in Figure 6 was given to the participants.

4.1. Tasks

All participants conducted the tasks given on three different workflows. We used the two workflows mentioned in the previous section, and added a non-assisted workflow as the baseline whereby participants had to apply pattern mining techniques to a predefined event set which could not be further modified. The non-assisted workflow would be comparable to using a set of independent tools including the SPMF library. Since the values for minimum support affect the size of the output and the execution time of sequential pattern mining algorithms, we set a value so that the parametrisation of the algorithms would not be a confounding factor. The value we set was empirically tested for the frequent itemset algorithm and the dataset under evaluation beforehand to make sure that the output would yield as many results as possible in an acceptable computation time. The value for support was set to 2%, meaning that, at least, 197 episodes had to have a given pattern in common. The execution consistently took a maximum of 5–6 seconds. Since higher support values increase the performance of sequential pattern mining algorithms, this would be considered an upper bound execution time (for this dataset) as the support value was relatively low.

[6] `http://www.cs.manchester.ac.uk`

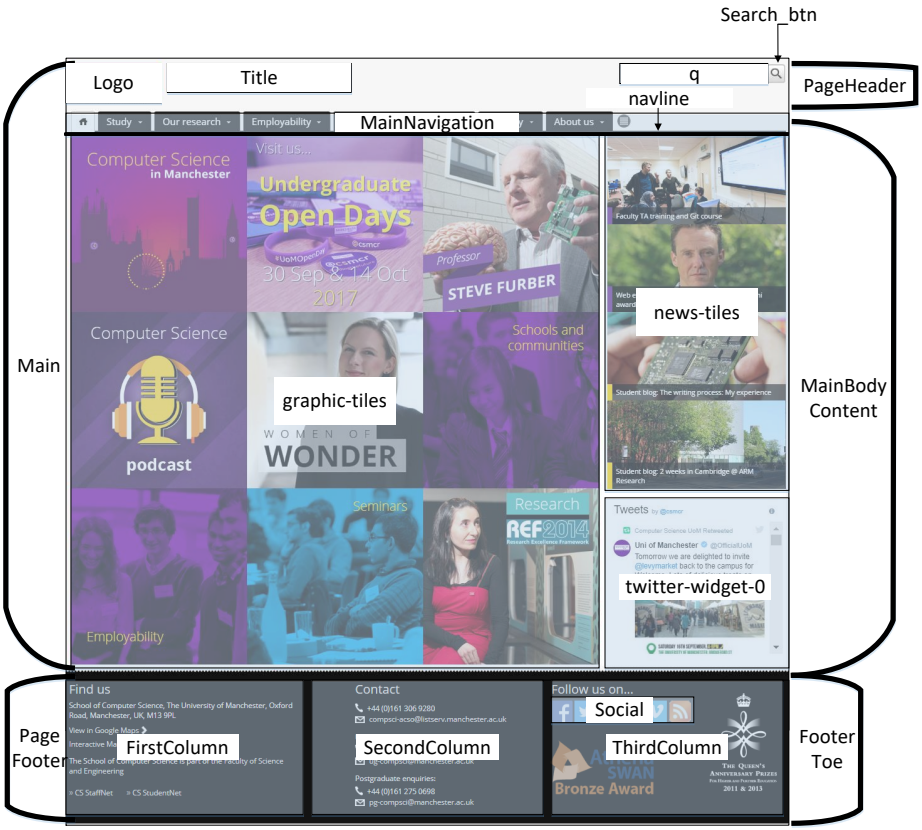

Figure 6: Printout of the Web page used for the study with annotations for the participants

In the assisted workflow, participants could modify and select a subset of the events to generate an event set that would be processed by the pattern mining algorithms. In addition to this, in the assisted++ workflow, users could also introduce hypotheses on the event set to narrow down the analysis. We acknowledge the difficulty of using WevQuery-PM for the first time, so we considered randomising the order. However, the assisted++ workflow necessarily builds on top of the assisted option, so participants would always carry out these tasks in the same order. Therefore, we only randomised the order between the non-assisted and the assisted workflows. For each option, we defined the following task types:

- **Guided** tasks where participants were given precise instructions to follow.

- **Exploration** task where participants were asked to interpret the results generated by the guided tasks.
- **Directed** tasks where participants were asked to carry out a task associated with the capabilities of the workflow used. Changing a parameter of the pattern mining algorithm in non-assisted workflow, modifying the input for the pattern mining algorithms in the assisted one, and user-created hypotheses in the assisted++ workflow.

After each task, participants filled in the component-based usability questionnaire (CBUQ) (Brinkman et al., 2009) to measure ease of use as well as the perceived difficulty using the perceived difficulty assessment questionnaire (PDAQ) (Ribeiro and Yarnal, 2010). The PDAQ was on a five-point Likert scale were '1' indicated "very difficult" and '5' meant "very easy". On completion of the study, participants filled in the USE usability questionnaire (Lund, 2001) where both assisted workflows were evaluated together. The ease of use (CBUQ) and usability (USE) questionnaires were on a five-point Likert scale where '1' indicated "strongly disagree" and '5' was for "strongly agree". Effectiveness scores and completion times for each task were jotted down on-the-fly using a timer. Additionally, we recorded the screen and the audio.

4.2. Procedure

Participants were told their goal was to obtain insights into the users' interaction by analysing the user interaction logs from the homepage of the website under evaluation. Participants were not expected to be familiar with the website so the manual of the study contained a full page (A4 size) coloured screenshot of the home page and a user manual defining the Web interaction events they had to deal with. On the screenshot we highlighted the most relevant components of the user interface, along with their ID attribute as indicated in the HTML source of the site. Participants were then able to use this screenshot to locate and associate ID names with components of the user interface, which was especially useful for non-self-explanatory IDs such as `q`, which was a text input field for the "search" functionality. Most of the IDs were self explanatory:

`pageHeader` and `footer` were areas of the page, `search_btn` was the button that submitted the search query typed in `q`, `title` was the title of the Web page that linked to the homepage, and `google_maps` was a link to open Google Maps on a specific location. We also made sure that the participants understood the concepts of "ID" and "node type", the user interaction events to be observed in the analysis, and the basics to interpreting the output of the pattern mining algorithms.

5. Results

Median completion times in directed tasks were 60 seconds (SD = 36) on the non-assisted workflow, 143 seconds on the assisted workflow (SD = 118) and 580 on the assisted++ workflow (SD = 235). Exploration tasks took a median time of 281 (SD = 149) seconds on the non-assisted, whereas accomplishing the exploratory tasks took users 290 (SD = 121) and 371 seconds (SD = 100) for the assisted and assisted++ workflow respectively. Longer completion times are observed in the exploratory tasks and the assisted workflows, which is confirmed by a one-way repeated-measures ANOVA, showing an effect of task on completion times $F(5,95) = 44.09$, $p < 0.0001$. A post-hoc Tukey test indicates significant differences[7] on the directed tasks between the non-assisted and assisted ($p < 0.03$), non-assisted and assisted++ ($p < 0.0001$), and assisted and assisted++ ($p < 0.0001$).

5.1. Usability

When we compare the baseline and the two assisted workflows (assisted and assisted++) the USE questionnaire yields medians of 3.7 (SD = 0.66) and 3.2 (SD = 0.67) for ease of use on the non-assisted and the assisted workflows respectively, 4.1 (SD = 0.57) and 3.75 (SD = 0.85) for ease of leaning, 3.6 (SD = 0.56) and 3.7 (SD = 0.46) for satisfaction and 3.6 (SD = 0.63) and 3.8 (SD =

[7] We do not report interactions between exploratory and directed tasks as they are of a different nature.

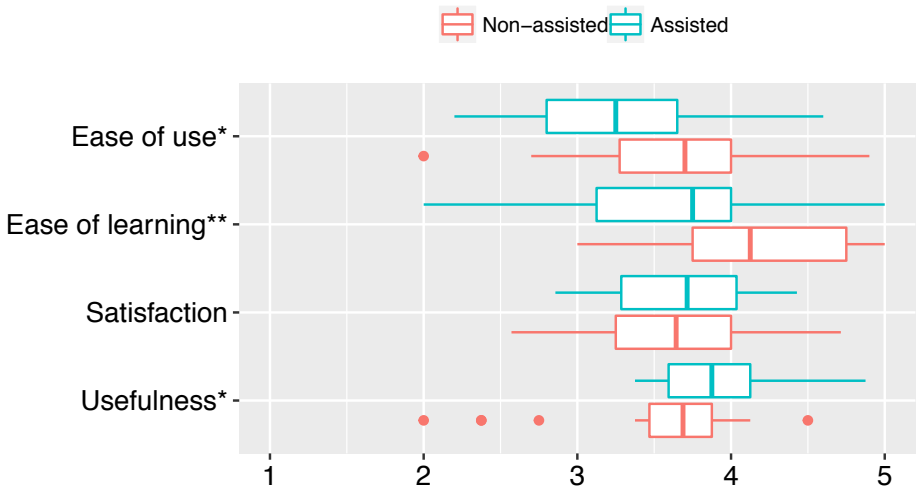

Figure 7: USE questionnaire: ease of use, ease of learning, satisfaction and usefulness of the workflows. Significance levels at *: $p < 0.05$, ** $p < 0.01$

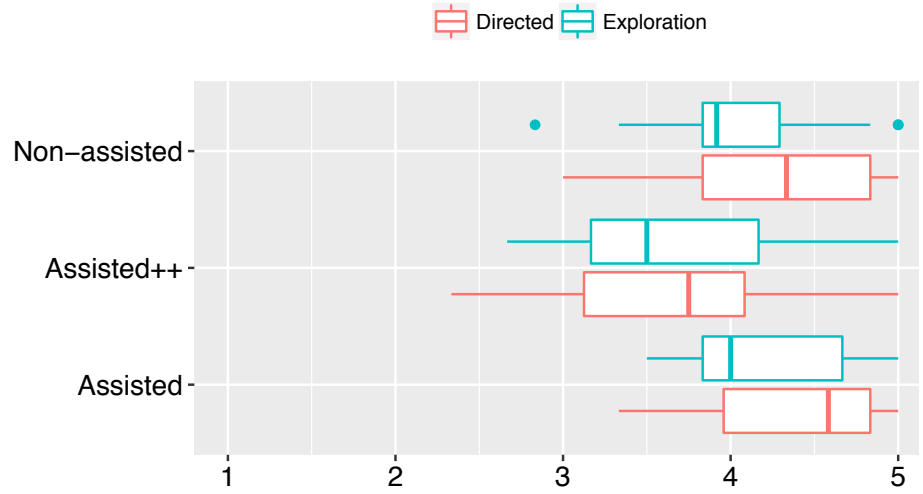

Figure 8: CBUQ questionnaire: ease of use of directed and exploratory tasks on the non-assisted and assisted workflows

0.40) for usefulness. Paired t-tests on these usability qualities yields significant differences for usefulness ($t(19) = -2.20$, $p < 0.05$), ease of use ($t(19) = 2.14$, $p < 0.05$) and highly significant differences for ease of learning ($t(19) = 3.84$, $p < 0.01$) —see the distribution of values in Figure 7.

In directed tasks, the CBUQ questionnaire for ease of use yields medians of 4.3 (SD = 0.56) on non-assisted tasks, 4.5 (SD = 0.56) on assisted tasks and 3.75 (SD = 0.80) on assisted++. As far as exploratory tasks are concerned, non-assisted tasks yield medians of 3.8 (SD = 0.56) and 4 (SD = 0.50) for assisted tasks and 3.5 (SD = 0.72) for assisted++. The boxplots in Figure 8 display the distribution of the values. A one-way repeated-measures ANOVA found a significant effect of type of task on ease of use, $F(5,95) = 8.22$, $p < 0.001$. Post-hoc Tukey tests show significant differences between the assisted and assisted++ workflows on exploratory ($p < 0.01$) and directed tasks ($p < 0.001$). On directed tasks differences are significant between the non-assisted and assisted++ workflow.

All tasks get a median of 4 (i.e. *easy*) for perceived difficulty as measured with the PDAQ questionnaire except for those tasks executed in the assisted++ workflow, which yield a median of 3 (i.e. *fair*). There is again an effect of task on difficulty, as indicated by a one-way repeated-measures ANOVA, $F(5,95) = 8.96$, $p < 0.0001$. A post-hoc Tukey test indicates significant differences between the assisted++ and assisted workflow, and assisted++ and non-assisted workflow on directed tasks ($p < 0.0001$). Marginally significant differences are found ($p = 0.08$) between the two assisted workflows on exploratory tasks.

5.2. Knowledge Discovery

Table 3 shows the discoveries made by the participants grouped by the workflow and the type of discovery: whether it was descriptive knowledge, inferred knowledge or the participant refined the current hypothesis. The types of discoveries map approximately to the learning objectives in Bloom's taxonomy for learning (Bloom et al., 1956): comprehension, analysis and synthesis. While we acknowledge other approaches to classify discoveries (Livingston et al., 2001),

our classification contemplates the formulation of new hypotheses.

Descriptive discoveries indicate a basic level of understanding of the output of the pattern mining algorithm, and users being able to distinguish the relevance of a pattern based on its frequency. Discoveries by *inference* suggest that the participants established links between the output of the pattern mining algorithms and particular behaviours exhibited on the Web page: e.g. clicks on the search text field might indicate that users are intending to use search functionalities. *Prospective hypotheses* were formulated when participants gave possible explanations for the discovered behaviours, which could be used to guide the creation of new hypotheses to be then reintroduced into the analysis workflows. Participants came up with 100 instances of discoveries that corresponded to the descriptive category, 65 instances belonging to inference and 21 prospective hypotheses were formulated.

In the non-assisted workflow, the explored event set included the occurrence of all the available combinations of events and contexts. Out of 51 discoveries, 38 belonged to the descriptive class, 10 to inference and 3 to prospective hypotheses. Participants were able to understand the output from the pattern mining but struggled to infer meaning from it: nine participants were able to recognise the top of the page as the main point of interest for users after identifying the interface elements that got most interactions, and one participant realised small interface elements triggered greater number of hover events. Another participant realised that due to the nature of the page (a homepage providing access to other parts of the website) a mouse click would typically indicate the end of the interaction, leading the user to a different Web page. This participant inferred the existence of intense mouse hovering activities would commonly take place before that click. Another participant hypothesised that users were trying to access navigation menus, while a last one assumed users were just exploring the Web page.

In the assisted workflow, out of 80 discoveries, 35 were of a descriptive nature, while 35 and 10 belonged to the inference and prospective hypothesis classes respectively. In this case, the event set was filtered by selecting mouse click

Table 3: Table of discovered knowledge. Participants made 100 discoveries belonging to the "Descriptive" category, 65 to the "Inference" category, and 21 to the "Prospective hypothesis" category

	Descriptive (100)	**Inference** (65)	**Prospective hypothesis** (21)
Non-assisted (51)	• Hovering `navline` is frequent (14) • Hovering `pageHeader` is frequent (10) • `mouseover` is the most frequent event (8) • Hovering `image` is frequent (3) • Hovering `graphic_tiles` is frequent (2) • Hovering `title` is frequent (1)	• Most of the interaction is at the top (4) • Menus at the top are used more (3) • The header of the page is used more (2) • Small items trigger more hover events (1)	• People exhibit hovering action before clicking (1) • Users are trying to access navigation menu (1) • Users are exploring the page (1)
Assisted (80)	• Interaction with `q` is relevant (19) • Clicks on `title` are frequent (8) • Clicks on `footer` are frequent (4) • Clicks on `google_maps` are frequent (3) • Clicks on `contact` are not frequent (1)	• `q` element is relevant, users are searching (16), and as `q` and `search_btn` are connected, they are using the search dialog (10) • Users click on `title` to go to homepage (4) • Users look for the school's location (2) • Users look for contact information (1) • Users might copy text from `footer` (1) • Users do not look for the email address (1)	• `q` and `search_btn` are only connected sometimes: users might be using the "enter" key (6), or it might indicate that users are not actually carrying out the search (2) • People searching might indicate that they cannot find what they want at first, does this indicate bad design? (2)
Assisted++ (55)	• Notice frequency of `hypothesis1` (19) • Clicks on `link` are frequent (7) • Clicks on `image` are frequent (1)	• Noticed the relation between `hypothesis1` and clicking on `link` (13), or other element (1) • Noticed several `hypothesis1` taking place in the same episode: users might be reading something (2), users might be waiting for the page to do something (2), there might be particularly slow users (1), or the interface might not be intuitive enough (1)	• Users might hover an element to disclose information to then click on a disclosed `link` (4) • Repeated hovering behaviour (`hypothesis1`) followed by a click on `link` (3). It might Indicate users have been exploring several menus till they found the information they were looking for (1)

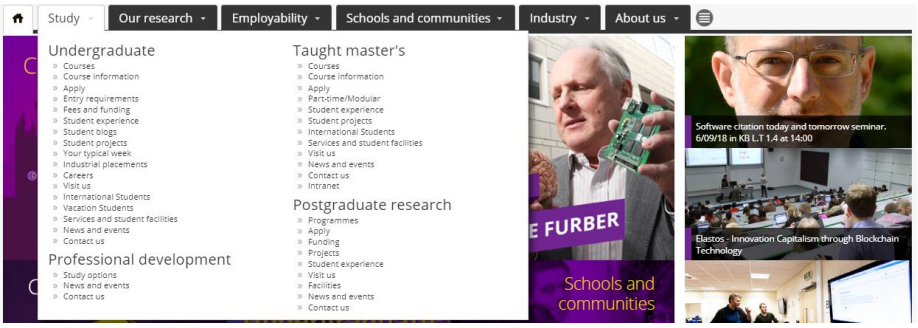

Figure 9: Example of interaction when hovering one of the menus in the Web page

events on interface elements with a known ID, which drastically reduced the size of the output. Nineteen participants immediately noticed the high frequency of interactions with the element that had `q` as an ID. This element was a text input field to type the keyword to conduct searches on the website. Based on this information, half of the participants were able to link the mouse interaction on the `q` input field with the `search button` element (a button next to the mentioned text input field, that submits the query and triggers the search) and determined that users were using the "search" feature that was available on the Web page. Noticeable but less frequent interactions with other interface elements were also identified, such as the `title` and the `footer` elements of the Web page. The interpretation of the role of the remaining user interface elements was generally speculative (e.g. users clicking on `title` to go to the homepage). Eight participants realised that only a subset of clicks on the `q` text input field took place together with `search button`, and formed prospective hypotheses. Six participants suggested adding keyboard interactions to the analysis (e.g. *"maybe they just press "enter" after writing something in* `q`*"*), and two participants suggested that users were intending to search, but changed their mind afterwards. Finally, two participants considered the use of the search function as an indicator of bad design of the homepage: *"Users are not finding what they are looking for straight away. It is not visible or easy to find"*.

In the assisted++ workflow, participants incorporated `hypothesis1`, as de-

fined in Table 2, into the pattern mining analysis workflow. 27 discoveries were descriptive, 20 were inferred and 8 were prospective hypotheses, accounting for a total of 55 discoveries. The analysed Web page contained interactive elements that disclose further information when hovered. For example, the main navigation bar contained a series of drop-down elements that, when hovered, disclosed a list of up to 45 links at once (see example in Figure 9). This hypothesis is shaped by a hypothetical user researcher's expectations and prior knowledge about the Web page under study: some users hovered these interactive elements for longer than 10 seconds, which is an abnormal behaviour worth exploring further. From the nineteen participants who explored the occurrences of `hypothesis1`, thirteen of them learned the relationship between `hypothesis1` and hyperlinks (defined as `link` elements in our system). Six participants also noticed multiple occurrences of `hypothesis1` within the same session and proposed possible explanations such as users reading and the existence of potential usability problems. Four participants took into account the nature of the analysed page and suggested that users could have been exploring the interactive elements, to then click on a link. Three participants pointed out repeated hovering activities before clicking on a link, and one of them suggested that the multiple occurrences of `hypothesis1` could also indicate that users had to traverse more than one menu (in a hierarchical menu) before finding the information they were looking for.

6. Discussion

While participants' completion times were higher on the assisted workflows, it did not have any negative effect on their effectiveness (i.e. whether the task was completed). In the assisted workflows, having tool support entailed being able to do more. This might have led to tasks that were perceived to be more difficult although this difficulty may not be related to cognitive complexity, but to having to do more. Despite being more difficult to learn, assisted workflows were found to be significantly more useful. There are two important factors to

take into consideration: first, the non-assisted baseline was designed as workflow that incorporated the algorithms of the SPMF library so that the comparisons of the usability of different workflows were *fair*. As such, selecting the input, the corresponding algorithm, establishing parameters and the preview of the pattern mining output was straightforward. Second, none of the participants had used WevQuery-PM before this study. Hence, additional functionalities such as the "Query creation" view of the assisted workflows affected, understandably, the ease of learning. The increased usefulness of the assisted workflows is encouraging in that it suggests that the user interface allowed to accomplish tasks of a complex nature. Therefore the lower perceived ease of use and higher difficulty of the assisted workflows —which is especially significant for the directed tasks— would be understandable and supports the idea that the introduction of extra functionalities to accomplish harder tasks was not detrimental but significantly beneficial from an utilitarian perspective.

Participants did not only regard the assisted workflows to be more useful, but their perception of usefulness was also empirically supported by the objective amount of actionable knowledge they acquired through these workflows. In the case of the non-assisted baseline, participants were capable of interpreting the output, and link it to particular behaviours exhibited on the Web page although only one participant formulated prospective hypotheses. It is worth highlighting that *just* by including pattern mining functionalities into the workflows enables users to acquire insights. As far as the assisted workflow is concerned, participants not only recognised particular behaviours (making possible to filter the event set using the identifiers of the interface elements), but also formed prospective hypotheses that could be reintroduced into the analysis pipeline. For example, many participants proposed including keyboard interaction to test the hypothesis that users were using the "enter" key to trigger the search action on the Web page. Other prospective hypotheses, such as the possibility of the Web page having a "bad design", could be considered weaker, as participants could not express how such condition could be tested. When it comes to the assisted++ workflow, the formulated hypotheses focused on be-

haviours that describe the complex process of exploring a Web page with a high information density. These hypotheses could have been transformed into queries by reformulating `hypothesis1` so that only those interface elements disclosing additional content are included (i.e. drop-down menues), and reintroduced into the analysis workflow for further exploration. It is worth highlighting that some of the above-mentioned insights might be usability problems. While the purpose of the proposed workflows is to provide support for a better understanding of the interactive behaviour of users on the Web, we acknowledge some of these behaviours might well be behavioural markers of underlying usability smells. Other behaviours might be just users realising their tasks as expected.

In summary, the basic assisted workflow provides the means to formulate a larger number of actionable knowledge, by allowing designers to tweak the original event set to produce domain-specific actionable knowledge and iteratively narrow down the produced results based on the acquired insights. This addresses *Challenge 1: cardinality, Challenge 2: semantics* and *Challenge 3: noise* discussed in the introductory section. In addition to these challenges, the assisted++ workflow enables users to introduce hypotheses to focus on particular behaviours and alleviate the process of filtering the results (and address *Challenge 4: complexity*).

Our results indicate that, if tools to reduce the complexities of data wrangling and pattern mining are provided, individuals who are knowledgeable about human factors on the Web could apply pattern mining techniques in their daily tasks. The consequences are noteworthy in that they open up opportunities to work with rich data and acquire insights about the use of interactive artefacts that could be incorporated in an iterative design process. Ultimately, this enables broader adoption of data-driven techniques applied to usability evaluation.

Limitations of WevQuery-PM and Threats to Validity. Our approach relies on matching queries against HTML elements including IDs, tag names and classes. XPath selectors would be, in principle, a more flexible alternative. Both approaches have strengths and weaknesses: while XPaths could target any element

on a website by default, WevQuery-PM requires to manually annotate elements with IDs when other attributes are absent. Yet, XPaths are dependent on the DOM so updates to the website would make this approach less sustainable over time. Our approach resists better structural updates at the cost of manual annotations and limited backwards compatibility. We acknowledge this approach is not exempt from updates either as `class` elements are mostly used for styling.

The number of pattern mining algorithms keeps growing and consequently SPMF, the pattern mining library deployed in our system, has added algorithms as late of this year. We narrowed down the use of pattern mining to one algorithm, which was found to be suitable for the analysed dataset. We did not give further details of the algorithms that were available in order to simplify the workflows.

Several participants suggested using visualisations to facilitate the interpretation of the resulting patterns. Since our goal was to evaluate the trade-off between including complex functionalities and the usefulness of the discoveries made, we decided not to factor visualisations in this investigation. Nevertheless, we acknowledge the advantages of using visualisations to show the pattern mining outputs, as well as other ranking techniques mentioned in the related work. Now that we have empirical evidence about the superiority of assisted workflows, future work will explore the incorporation of visualisations as a way to reduce the complexity. We discuss the threats to validity in our study:

- Sample of participants. Our sample represented user researchers who were experienced in Web technologies and were knowledgeable about human factors on the Web but lacked the expertise to use pattern mining tools. Whether WevQuery-PM can support data-savvy user researchers or data scientists without UX experience we cannot tell. We suspect that the former group already use self-tailored workflows, while the latter group may be able to use WevQuery-PM after a training period.

- Familiarity with the topic under evaluation. Being familiar with the domain of the content under evaluation is an important aspect to interpret

pattern mining outputs (Dev and Liu, 2017). None of the participants had accessed the Web page under study before although they were provided with a manual containing the key elements of the user interface. Yet, the website belonged to a higher education institution whose contents would not be completely unfamiliar to participants as all of them had engaged in higher education programmes before. If there was any lack of familiarity with the domain, this did not prevent participants from carrying out the tasks and making discoveries.

- Representativeness of the homepage. It is well documented that the homepage is where developers and designers spend most of their efforts (Nielsen and Tahir, 2001) so we acknowledge that the discoveries we report may not represent all the behaviours exhibited on the entire website although we would expect a significant overlap.

7. Future Research Avenues

Our results inform design recommendations that could be incorporated by systems including functionalities for Web log data wrangling and mining. We also identify research avenues and opportunities for future work.

Further automatisation for data processing. Existing approaches have focused on the segmentation of demographics found in common Web analytic tools, the splitting of event sequences (Law et al., 2018), or changing the level of detail to display (Perer and Wang, 2014). All these approaches identified possible barriers that prevented agile iterations. In our case, the extraction and cleansing of interaction events and behaviours were addressed by the proposed workflows.

The parametrisation of pattern mining algorithms (i.e. support) determine the number of patterns. This often requires to follow trial and error strategies to find the output that is more manageable (in terms of size) and semantically meaningful. Tool support to find the sweet spot will be of great help to the users of such systems.

Identification of usability smells. In the context of our work, frequent interaction patterns could be attributed either to expected behaviours or to systematic usability problems. In WevQuery-PM, functionalities to formulate hypotheses support user researchers as to whether certain behavioural patterns are indicative of usability problems. The same rationale applies when dealing with outlying behaviours, which may fall under the categories of *noise*, sophisticated strategies or problematic interactions. While experience, training and domain knowledge help in distinguishing usability problems from expected behaviours, a catalogue of generalisable problematic interactions (as informed by the literature (Paternò et al., 2017)) that could be matched against the interaction patterns found could a the first step.

Hypothesis formulation using natural language. Understandably, the functionalities provided to formulate hypotheses in the assisted++ workflow increased the perceived complexity of the task. While this was not detrimental to accomplishing the task itself, it suggests that other alternative ways of expressing hypothesis could be explored. Using controlled natural languages may remove barriers, especially if it is combined with auto-suggest functionalities.

8. Conclusion

We propose a set of functionalities to reduce the barriers that prevent user researchers from incorporating pattern mining algorithms in the analysis of interactive behaviours on the Web. To do so, we identify the requirements needed to address such challenges and provide two tool-supported workflows to (i) transform the input raw data to facilitate the exploration of interaction data; (ii) tackle the noise generated by pattern mining algorithms; and (iii) define complex interactive behaviours to identify regularities, potential usability problems and outlying behaviours. These workflows enable agile analyses, where user researchers can shape their insights as hypotheses, which can be refined iteratively.

We found that user researchers can discover actionable knowledge from low-level Web log data provided that functionalities for data wrangling and data mining remove the complexity around these tasks. Our study suggests that while a baseline system does not prevent this from happening, tool support (assisted workflows) facilitates higher order knowledge discoveries. The perceived difficulty of the assisted workflows is counterbalanced by both the perceived usefulness and the higher number of actionable knowledge discoveries.

Acknowledgements

This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreement H2020-693092 MOVING `http://moving-project.eu` and the EPSRC [EP/I028099/1].

References

Agrawal, R., Srikant, R., others, 1994. Fast algorithms for mining association rules, in: Proc. 20th int. conf. very large data bases, VLDB, pp. 487–499.

Akers, D., Simpson, M., Jeffries, R., Winograd, T., 2009. Undo and erase events as indicators of usability problems, in: Proceedings of the 27th international conference on Human factors in computing systems, ACM. pp. 659–668.

Apaolaza, A., Harper, S., Jay, C., 2013. Understanding users in the wild, in: Proc. of the 10th International Cross-Disciplinary Conference on Web Accessibility, pp. 13:1–13:4.

Apaolaza, A., Vigo, M., 2017. WevQuery: Testing hypotheses about web interaction patterns 1, 4:1–4:17.

Bloom, B.S., Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R., 1956. Taxonomy of educational objectives, handbook I: The cognitive domain. volume 19. New York: David McKay Co Inc.

Borgelt, C., 2012. Frequent item set mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2, 437–456.

Breslav, S., Khan, A., Hornbæk, K., 2014. Mimic: Visual analytics of online micro-interactions, in: Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, ACM. pp. 245–252.

Brinkman, W.P., Haakma, R., Bouwhuis, D.G., 2009. The theoretical foundation and validity of a component-based usability questionnaire. Behav. Inf. Technol. 28, 121–137.

Burg, B., Bailey, R., Ko, A.J., Ernst, M.D., 2013. Interactive record/replay for web application debugging, in: Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology, ACM, New York, NY, USA. pp. 473–484.

Carta, T., Paternò, F., de Santana, V.F., 2011. Web usability probe: A tool for supporting remote usability evaluation of web sites, in: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (Eds.), Human-Computer Interaction – INTERACT 2011, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 349–357.

Chudá, D., Krátky, P., Burda, K., 2018. Biometric Properties of Mouse Interaction Features on the Web. Interacting with Computers 30, 359–377. `http://oup.prod.sis.lan/iwc/article-pdf/30/5/359/25998731/iwy015.pdf`.

Dean, J., Ghemawat, S., 2008. Mapreduce: Simplified data processing on large clusters. Commun. ACM 51, 107–113.

Deka, B., Huang, Z., Franzen, C., Nichols, J., Li, Y., Kumar, R., 2017. Zipt: Zero-integration performance testing of mobile app designs, in: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, ACM, New York, NY, USA. pp. 727–736.

Deka, B., Huang, Z., Kumar, R., 2016. Erica: Interaction mining mobile apps, in: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, ACM, New York, NY, USA. pp. 767–776.

Dev, H., Liu, Z., 2017. Identifying frequent user tasks from application logs, in: Proceedings of the 22Nd International Conference on Intelligent User Interfaces, ACM. pp. 263–273.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., 1996. Advances in knowledge discovery and data mining, American Association for Artificial Intelligence, Menlo Park, CA, USA. chapter From Data Mining to Knowledge Discovery: An Overview, pp. 1–34.

Fournier-Viger, P., Lin, J.C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., Lam, H.T., 2016. The SPMF open-source data mining library version 2, in: Machine Learning and Knowledge Discovery in Databases, Springer, Cham. pp. 36–40.

Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M., 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth, in: proceedings of the 17th international conference on data engineering, pp. 215–224.

Heer, J., Chi, E.H., 2002. Separating the swarm: Categorization methods for user sessions on the web, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA. pp. 243–250.

Hilbert, D.M., Redmiles, D.F., 2000. Extracting usability information from user interface events. ACM Comput. Surv. 32, 384–421.

Law, P., Liu, Z., Malik, S., Basole, R.C., 2018. Maqui: Interweaving queries and pattern mining for recursive event sequence exploration. IEEE Transactions on Visualization and Computer Graphics , 1–1.

Lewis, C., 1982. Using the 'thinking-aloud'm ethod in cognitive interface design. Research Report RC9265, IBM TJ Watson Research Center .

Li, W., Harrold, M.J., Görg, C., 2010. Detecting user-visible failures in AJAX web applications by analyzing users' interaction behaviors, in: Proceedings of the IEEE/ACM international conference on Automated software engineering, ACM. pp. 155–158.

Liu, Z., Kerr, B., Dontcheva, M., Grover, J., Hoffman, M., Wilson, A., 2017. Coreflow: Extracting and visualizing branching patterns from event sequences. Computer Graphics Forum 36, 527–538.

Livingston, G.R., Rosenberg, J.M., Buchanan, B.G., 2001. Closing the loop: An agenda-and justification-based framework for selecting the next discovery task to perform, in: Data Mining, 2001. ICDM 2001, Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE. pp. 385–392.

Lund, A.M., 2001. Measuring usability with the use questionnaire. Usability interface 8, 3–6.

Malik, S., Shneiderman, B., Du, F., Plaisant, C., Bjarnadottir, M., 2016. High-volume hypothesis testing: Systematic exploration of event sequence comparisons. ACM Trans. Interact. Intell. Syst. 6, 9:1–9:23.

Mannila, H., Toivonen, H., Verkamo, A.I., 1997. Discovery of frequent episodes in event sequences 1, 259–289.

Mooney, C.H., Roddick, J.F., 2013. Sequential pattern mining – approaches and algorithms. ACM Comput. Surv. 45, 19:1–19:39.

Nielsen, J., Tahir, M., 2001. Homepage Usability: 50 Websites Deconstructed. New Riders Publishing, Thousand Oaks, CA, USA.

Paternó, F., Schiavone, A., Pitardi, P., 2016. Timelines for mobile web usability evaluation, in: Proc. of the International Working Conference on Advanced Visual Interfaces, pp. 88–91.

Paternò, F., Schiavone, A.G., Conti, A., 2017. Customizable automatic detection of bad usability smells in mobile accessed web applications, in: Proceedings of the 19th International Conference on Human-Computer Interaction

with Mobile Devices and Services, ACM, New York, NY, USA. pp. 42:1–42:11.

Perer, A., Wang, F., 2014. Frequence: Interactive mining and visualization of temporal frequent event sequences, in: Proceedings of the 19th International Conference on Intelligent User Interfaces, ACM. pp. 153–162.

Pirolli, P., Card, S., 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis, in: Proceedings of international conference on intelligence analysis, pp. 2–4.

Ribeiro, N.F., Yarnal, C.M., 2010. The perceived difficulty assessment questionnaire (pdaq): Methodology and applications for leisure educators and practitioners. Schole 25.

Rzeszotarski, J., Kittur, A., 2012. Crowdscape: Interactively visualizing user behavior and output, in: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, ACM, New York, NY, USA. pp. 55–62.

de Santana, V.F., Baranauskas, M.C.C., 2015. Welfit: A remote evaluation tool for identifying web usage patterns through client-side logging. International Journal of Human-Computer Studies 76, 40 – 49.

Sarkar, A., Spott, M., Blackwell, A.F., Jamnik, M., 2016. Visual discovery and model-driven explanation of time series patterns, in: 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 78–86.

Seno, M., Karypis, G., 2002. SLPMiner: an algorithm for finding frequent sequential patterns using length-decreasing support constraint, in: 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pp. 418–425.

Silberschatz, A., Tuzhilin, A., 1995. On subjective measures of interestingness in knowledge discovery, in: Proceedings of the First International Conference on Knowledge Discovery and Data Mining, AAAI Press. pp. 275–281.

Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N., 2000. Web usage mining 1, 12.

Vigo, M., Harper, S., 2017. Real-time detection of navigation problems on the world 'wild' web. International Journal of Human-Computer Studies 101, 1 – 9.

Weinreich, H., Obendorf, H., Herder, E., Mayer, M., 2006. Off the beaten tracks: exploring three aspects of web navigation, in: Proceedings of the 15th international conference on World Wide Web, ACM. pp. 133–142.

Zgraggen, E., Drucker, S.M., Fisher, D., DeLine, R., 2015. (s,qu)eries: Visual regular expressions for querying and exploring event sequences, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM. pp. 2683–2692.

Zhang, X., Brown, H.F., Shankar, A., 2016. Data-driven personas: Constructing archetypal users with clickstreams and user telemetry, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA. pp. 5350–5359.

Zhao, J., Liu, Z., Dontcheva, M., Hertzmann, A., Wilson, A., 2015. MatrixWave: Visual comparison of event sequence data, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM. pp. 259–268.