Repetitive Event Patterns Search in User Activity Data

Nikita A. Danilov[1], Tatiana E. Shulga[2], Aleksandr A. Sytnik[3]
School of Applied Information Technology and Communication
Yuri Gagarin State Technical University of Saratov, Saratov, Russia
[1]Nikita_Danilov@outlook.com, [2]shulga@sstu.ru, [3]as@sstu.ru

***Abstract*— The paper describes the approaches used in usability assessing of user interfaces based on the idea of searching patterns in user behavior. The existing researches in this direction are described. It is shown that the task of searching patterns of user behavior can be represented as a task of searching for sequential patterns of data and apply appropriate methods.**

Keywords*— *usability; user interface; repetitive patterns; mathematical methods; user activity data; sequential patterns

I. Overview of Existing Approaches

The software user formulates a plan in his head and performs it as a sequence of actions [1] to achieve the goal. After that, he examines the state of the interface and evaluates the result, determining what actions it needs to perform further in accordance with the previously set goal.

Difficulties in the interaction between user and software interface may arise due to the complexity of the action plan formulation (decision on further action, articulation translation) or lack of understanding of the system response (the relationship of changes in the interface with the actions performed, observation translation) [1]. These issues affect user behavior. For example, there is a pause in the process of interaction to select further actions. Or, the user tries different actions in a row, returning after receiving an incorrect result. The presence of such sequences of actions (event patterns) can be associated with usability problems of the user interface.

User activity data contains a lot of information about the process of user interaction with the software interface [2]. Quite a long time ago, researchers realized that it is possible to automatically analyze the collected data to find similar patterns of events that signal problems of usability [1]. Automatic analysis saves time and money, because experts focus on specific areas where such patterns have been identified, instead of analyzing all the data.

An indicator of the usability problems is the presence of frequently repeated identical sequences of actions (repetitive pattern) [1]. That can mean that the user is trying to reach the goal and fails each time. However, it’s noted that repetitive patterns can also be formed due to repetitive tasks, which can be normal correct behavior. Therefore, we need to understand the specifics of a software.

Some researchers suggest tracking such metrics (as indicators of possible usability problems) [4]: the number of online help calls; the number of undo actions; the number of error messages or warnings encountered; the number of actions without the expected effect; frequent opening-closing of drop-down lists; pressing the same button more than once.

Others are based on the detection of information search problems by the user, while browsing the website [5]. For example, a pattern associated with the user's mouse movement is highlighted. It’s known that users use the mouse cursor not only to click links or buttons. In the process of visual search on a page, the user typically moves the cursor after the items. Since elements are often located horizontally or vertically, the user moves the mouse cursor horizontally or vertically, respectively. Therefore, it is possible to detect the vertical/horizontal mouse cursor movement event pattern. This is an indicator that in this case the user has spent a lot of time searching for the item.

Stable scrolling events on the page, with almost the same pauses, indicate that the user is reading the content. However, a series of quick up-and-down scrolling actions may mean that the user is forced to scan the entire contents of the page trying to find the right block of information on the page [5].

Sandrine Balbo (Sandrine Balbo) identified similar patterns of events more detailed and developed the software WAUTER (Web Automatic Usability Testing EnviRonment) [3]. She proposed a task-oriented model for evaluating the usability of the software system. User actions are compared to an ideal task model to detect any inconsistencies between them. In this case, the following patterns are highlighted:

- Action Cancellation. Cancel the action as soon as it is executed. For example, a user returns to the previous web page immediately after navigating to it. A possible reason is that the user did not plan to visit this web page and moved by accident, due to navigation problems.
- Action Re-occurrence. Frequent repetitions of simple actions, such as mouse clicks or keystrokes, may indicate insufficient responsiveness of the interface, which leads the user to think that the system does not recognize its action.

978-1-5386-5824-6/18/$31.00 ©2018 IEEE

- Direction Shift. Change direction when the user stops the current task and starts another. For example, a user can stop searching for an item in a list and instead go to the search in a text box. A possible cause is the difficulty of limiting further action.
- Irritant Actions. Unrelated actions in the process of the task also signal usability problems.

Some researchers identify patterns associated with viewing the website and transitions between web pages (browsing) [6]. The user's move paths are compared to some set of expected and well-known paths. Different ways belong to the category of unexpected (unexpected browsing) and can signal about the problems of ease of use.

A pattern associated with transitions between web pages can also be a situation where the user quickly moves from page to page without staying on each [5].

II. SEARCH OF EVENT PATTERNS

These methods of user behavior patterns search have much in common with the task of search sequential patterns of Data Mining [7]. In most cases, all patterns are sequential and only events kinds vary. However, user activity data is almost always not a set of short transactions, but a large set of actions, which in most cases cannot be correctly divided into subsets [3].

As noted above, one of the possible reasons for the occurrence of regularly repeated patterns in user activity data is the presence of errors or difficulties in interaction with the interface. In this case, there may be a decrease in both the efficiency and effectiveness of users. Therefore, reducing the number of such patterns reduces the risk of errors.

Another possible reason for having repetitive patterns in user activity data is the need to perform the same recurring action chains to complete the tasks. Naturally, automation interim action reduces resource costs. Therefore, the less the user performs the same type of chain of actions, the less risk of error and less resource costs, and therefore, the more effective the interaction.

Let’s introduce a few basic concepts necessary to adapt the theory of sequential pattern mining to the field of usability evaluation.

Event – a fact recorded at a certain time during the interaction of a certain user on a certain device with the software user interface [11]. An event has a non-empty unique set of attributes: registration time, event type (for example, mouse click, command execution), and special attributes that depend on the event type. For example, a mouse click event can contain: the name of the key (left, right), the type of click (single, double), the coordinates of the mouse cursor position, etc. Command execution event (command event) can contain: button name, command name, execution method (mouse click, hot key, “Enter” key).

Denote $E = \langle e_i \rangle$, i=1,n – set of all events ordered by the time of registration. It should be noted that each event has a non-empty unique set of attributes, and, therefore, is unique.

A. Events classification

In a set without repetitions, it is impossible to have repetitive patterns as ordered subsets. Therefore, a preliminary classification of all events is required.

Denote $C = \{c_i\}$, i=1,k as the set of defined event classes. The class of events is an arbitrary set of events that have a certain property or feature. For example, expert can classify events by type, by the name of the executed command (for command events), by the user interface area (in the top menu area, in the main workspace).

To classify events, we need a multi-valued classifying function *f*, which can be written as: $f(e) \subset C : f(e) = \langle c \in C \rangle$.

The reason why the function is multi-valued comes from the variety of possible research directions of user activity data set. The classification rules and the set of classes are determined by the expert. Depending on the goals of the study, some events can be ignored. For example, an expert may be interested only in command events, i.e. those related to the call of a command. In this case, we will say that the filtering of other types of events is performed. On the other hand, it is possible to map several event classes to one event, in case of a special set of attributes.

In the further, the events will be understood as the events after the classification, because the search for sequential patterns is impossible to these actions because of the uniqueness of each event.

B. Event patterns

The theory of sequential patterns implies that there are many candidates for whom the level of support is checked. We define the concepts of pattern and support for a set of events.

Denote $P = \{p_i\}$, i=1,r – set of candidates of sequential event patterns. The pattern is the arrangement of elements of the set *C* with repetitions: pi = $\langle c_{i1} \ldots c_{ij} \rangle$, where j = 1,$q_i$ is the power of p_i placement, i.e. the number of events in the pattern. We will count the number of events in the specified location as the pattern length. We assume that the pattern *p* is included in the set of events *E* if all elements *p* are contained in *E*, and the order of elements in a subset of *E* corresponds to the order of elements p.

We denote μ_p as the number of occurrences of the pattern *p* in the set *E*, calculated as the number of disjoint ordered occurrences. For example, suppose we have events $\langle \mathbf{2, 4, 2}, 4, 2, 1, 2 \rangle$ and the pattern $\langle 2, 4, 2 \rangle$. The number of occurrences will be equal to 1, since the 4th element was already present during the counting.

The pattern *p* is called supported if the number of occurrences μ_p is greater than zero.

Let λ_p denote the support of pattern *p*, calculated as: $\lambda_p = (\mu_p * q) / n$, where μ_p is the number of occurrences of a pattern, *q* is the length of pattern *p*, *n* is the number of events.

Support takes a value on the interval [0, 1]. Thus, it’s possible to describe the value of support as a proportion of the content of the pattern in the events. This is necessary to compare different patterns of influence on the user's interaction with the software, which cannot be done based on the number of occurrences alone.

For example, suppose we have events: $\langle 1, 2, 3, 2, 1, 1, 2, 3, 2, 1\rangle$.

Calculate the values of μ and λ for the following patterns:

- $p_1 = \langle 1, 2\rangle$, $\mu = 2$, $\lambda = 0.4$;
- $p_2 = \langle 1, 2, 3\rangle$, $\mu = 2$, $\lambda = 0.6$;
- $p_3 = \langle 2, 3\rangle$, $\mu = 2$, $\lambda = 0.4$;
- $p_4 = \langle 1, 2, 3, 2, 1\rangle$, $\mu = 2$, $\lambda = 1$.

The number of occurrences of all patterns is equal to 2. However, the support of the p_4 pattern is equal to 1, i.e. it completely makes up the whole set of events, and therefore is a more likely candidate for the expert's close attention. Improving the efficiency of user interaction with the software interface in the affected by such a pattern can dramatically improve the efficiency of the entire process of working with the software.

C. Patterns analysis

Data Mining domain has a well-developed theory of searching for associative rules and sequential patterns. The adaptation and application of existing methods (such as AprioriAll [8], AprioriSome, DynamicSome, etc.) will significantly optimize the calculation of support values for generated pattern candidates.

Having support values for each pattern, the expert can focus on the most significant of them for the process of working with the software. The set of patterns will depend on the objectives of the analysis.

Further, the expert can hypothesize the necessary changes in the user interface to improve the efficiency of user interaction with the software. In making decisions, the expert must consider many different factors: software features, psychological factors of software use and user characteristics.

Changing the user interface will change the composition of events and sequential patterns, because the sequence of actions required to achieve the goals set by users will change.

After making changes to the program interface, it is possible to re-collect and analyze user activity data, which can confirm or refute the previous hypothesis.

III. CONCLUSION

The review shows that there is an extensive layer of research in the field of analysis of patterns of user behavior. This subject has a great potential, in view of the obvious savings of resources and time in the automatic analysis of data user activity to detect aspects of the software interface that require detailed research.

The task of repetitive event patterns search in user activity data has much in common with the task of sequential patterns search from the data mining domain. Therefore, it is possible to adapt and apply the existing theory, which will allow to use the previously created developments in the new field.

REFERENCES

[1] Siochi A.C., Ehrich R.W. Computer Analysis of User Interfaces Based on Repetition in Transcripts of User Sessions. ACM Transactions on Information Systems. 1991, vol. 9, no. 4, pp. 309-335.

[2] Danilov N., Shulga T., Frolova N., Melnikova N., Vagarina N., Pchelintseva E. Software usability evaluation based on the user pinpoint activity heat map. Advances in Intelligent Systems and Computing. 2016, vol. 465, pp. 217-225 (in Russian).

[3] Balbo S., Goschnick S., Tong D., Paris C. Leading Usability Evaluations to WAUTER. Proceedings of the 11th Australian World Wide Web Conference (AusWeb). Gold Coast, Australia, Southern Cross University. 2005, pp. 279-290.

[4] Swallow J., Hameluck D., Carey T. User Interface Instrumentation for Usability Analysis: A Case Study. CASCON ’97, Toronto, Ontario. November 1997. Case Study.

[5] Shah I. Event Patterns as Indicators of Usability Problems. Journal of King Saud University. Computer and Information Sciences. 2008, vol. 20, pp. 31-43,

[6] Ting I., Kimble C., Kudenco D. UBB Mining: Finding Unexpected Browsing Behaviour in Clickstream Data to Improve a Web Site’s Design. Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI2005). France. 2005. Pp. 179-185.

[7] Mabroukeh N.R., Ezeife C.I., A taxonomy of sequential pattern mining algorithms. ACM Computing Surveys (CSUR). 2010, vol. 43 (1), article no. 3.

[8] Aloysius G., Binu D. An Approach to Products Placement in Supermarkets Using PrefixSpan Algorithm. Journal of King Saud University-Computer and Information Sciences. 2013, vol. 25 (1), pp. 77–87.