

Поиск повторяющихся шаблонов событий в данных активности пользователей

Н. А. Данилов¹, Т. Э. Шульга², А. А. Сытник³

Саратовский государственный технический
университет им. Гагарина Ю.А

¹Nikita_Danilov@outlook.com, ²Shulga@sstu.ru, ³as@sstu.ru

Аннотация. В работе рассматриваются подходы, используемые при оценке удобства использования пользовательских интерфейсов, основанные на идее поиска закономерностей и шаблонов в поведении пользователей. Описываются существующие исследования в данном направлении. Показывается, что задачу поиска шаблонов поведения пользователей можно представить как задачу поиска последовательных шаблонов из области интеллектуального анализа данных (англ. Data Mining) и применять соответствующие методы.

Ключевые слова: удобство использования; пользовательский интерфейс; последовательные шаблоны; математические методы; данные активности пользователей

I. ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ

Пользователь программного обеспечения, для достижения поставленной цели, формулирует план в голове и выполняет его как последовательность действий [1]. После выполнения, он изучает состояние интерфейса и оценивает полученный результат, определяя какие действия ему необходимо выполнить дальше в соответствии с поставленной ранее целью.

Затруднения, при взаимодействии пользователей с программным интерфейсом, могут возникать по причине сложности формулирования плана действия (принятия решения о дальнейших действиях, англ. articulation translation) или непонимания ответа системы (взаимосвязь изменений в интерфейсе с выполненными действиями, англ. observation translation) [1]. Указанные проблемы влияют на поведение пользователя. Например, возникает пауза в процессе взаимодействия, для выбора дальнейших действий. Либо, пользователь пробует различные действия подряд, возвращаясь обратно после получения некорректного результата. Наличие подобных последовательностей действий (шаблонов событий) можно связывать с проблемами юзабилити пользовательского интерфейса.

Данные активности пользователей содержат огромное количество информации о процессе взаимодействия пользователей с программным интерфейсом [2]. Довольно давно исследователи поняли, что возможен автоматический анализ собираемых данных, с целью поиска подобных шаблонов событий, сигнализирующих о

проблемах удобства использования [1]. Автоматический анализ экономит время и деньги, т.к. эксперты, вместо анализа всех данных, фокусируют своё внимание на отдельных областях, где были выявлены подобные шаблоны.

Индикатором проблем удобства использования, является наличие часто повторяемых одинаковых последовательностей действий (англ. action repetition pattern) [1]. Это может означать, что пользователь пытается достичь цели и каждый раз терпит неудачу. Однако, отмечается что повторяемые шаблоны так же могут быть образованы из-за повторяющихся задач, которые могут быть нормальным корректным поведением. Поэтому, требуется понимание специфики конкретного программного обеспечения.

Отдельные исследователи, в качестве индикаторов возможных проблем удобства использования, предлагают отслеживать [4]: количество вызовов онлайн-справки; количество действий отмены; количество возникших сообщений об ошибке или предупреждений; количество действий без ожидаемого эффекта; частое открытие-закрытие выпадающих списков; нажатие одной и той же кнопки более одного раза.

Другие, основываются на обнаружении проблем поиска информации пользователем, в процессе просмотра веб-сайта [5]. Например, выделяется шаблон, связанный с движением мыши пользователем. Известно, что пользователи используют курсор мыши не только для нажатия ссылок или кнопок. В процессе визуального поиска на странице, пользователь обычно перемещает курсор вслед за элементами. Т.к. элементы часто расположены горизонтально или вертикально, то пользователь соответственно тоже перемещает курсор мыши горизонтально или вертикально. Следовательно, возможно обнаружение шаблона событий перемещения курсора мыши (англ. vertical/horizontal mouse movement). Это выступает индикатором, что в данном случае пользователь затратил много времени на поиск элемента.

Устойчивые события прокрутки (скроллинга, англ. scroll) на странице, с почти одинаковыми паузами, сигнализируют что пользователь читает содержимое. Однако, серия быстрых действий прокрутки вверх-вниз может означать что пользователь вынужден сканировать

всё содержимое страницы пытаюсь найти нужный блок информации на странице [5].

Сандрин Бальбо (Sandrine Balbo) более детально идентифицировала подобные шаблоны событий, создав программное обеспечение WAUTER (Web Automatic Usability Testing Environment) [3]. Она предложила ориентированную на задачи модель оценки удобства использования программной системы. Пользовательские действия сравниваются с идеальной моделью задач, для обнаружения любых несоответствий между ними. При этом, выделяются следующие шаблоны:

- Action Cancellation. Отмена действия сразу после его выполнения. Например, пользователь возвращается на предыдущую веб-страницу сразу же после перехода на неё. Возможной причиной является то, что пользователь не планировал посетить эту веб-страницу и перешел случайно, из-за проблем с навигацией.
- Action Re-occurrence. Частое повторение простых действий, таких как клики мыши или нажатия клавиш, может означать недостаточную отзывчивость интерфейса, которая ошибочно приводит пользователя к мысли что система не распознает его действие.
- Direction Shift. Смена направления, когда пользователь прекращает выполнения текущей задачи и приступает к другой. Например, пользователь может остановить поиск элемента в списке и вместо этого перейти к поиску в текстовом поле. Возможной причиной является наличие трудностей, ограничивающих выполнение дальнейших действий.
- Irrelevant Actions. Несвязанные действия в процессе выполнения задачи также сигнализируют о проблемах юзабилити.

Некоторые исследователи, выделяют шаблоны, связанные с просмотром веб-сайта и переходами между веб-страницами (англ. browsing) [6]. Пути перемещения пользователя сравниваются с некоторым установленным набором ожидаемых и общеизвестных путей. Отличающиеся пути относятся к категории неожиданных (unexpected browsing) и могут сигнализировать о проблемах удобства использования.

Шаблоном, связанным с переходами между веб-страницами, может являться и ситуация, когда пользователь быстро переходит со страницы на страницу, не задерживаясь на каждой [5].

II. ПОИСК ШАБЛОНОВ СОБЫТИЙ

Перечисленные методы поиска шаблонов поведения пользователей имеют много общего с задачей поиска последовательных шаблонов из области интеллектуального анализа данных (англ. Data Mining) [7]. В большинстве случаев все шаблоны являются последовательными, варьируется лишь то, какие именно события анализируются. Однако, данные активности

пользователей почти всегда представляют собой не множество коротких транзакций, а большие наборы действий, которые в большинстве случаев невозможно корректно разделить на поднаборы [3].

Как отмечалось выше, одной из возможных причин появления регулярно повторяющихся шаблонов в данных активности пользователей, является наличие ошибок или затруднений при взаимодействии с интерфейсом. В этом случае может наблюдаться снижение и результативности, и эффективности пользователей. Следовательно, уменьшение числа подобных шаблонов, уменьшает риск возникновения ошибок.

Другой возможной причиной наличия повторяющихся шаблонов в данных активности пользователей является потребность выполнения одних и тех же повторяющихся цепочек действий для выполнения поставленных задач. Закономерно, что автоматизация промежуточных действий уменьшает затраты ресурсов. Следовательно, чем меньше пользователь совершает однотипных цепочек действий, тем меньше риск допущения ошибки и меньше затраты ресурсов, а значит, тем эффективнее взаимодействие.

Введем несколько основных понятий, необходимых для адаптации теории поиска последовательных шаблонов применительно к области оценки удобства использования.

Событие – факт, зафиксированный в определенный момент времени при взаимодействии определенного пользователя на определенном устройстве с программным пользовательским интерфейсом [11]. Событие обладает непустым уникальным набором атрибутов: время регистрации, вид события (например, клик мыши, исполнение команды) и специальные атрибуты, зависящие от вида события. Например, событие клика мыши может содержать: наименование клавиши (левая, правая), тип клика (одинарный, двойной), координаты положения курсора мыши и т.д. Событие исполнения команды (командное событие) может содержать: наименование кнопки, наименование команды, способ исполнения (клик мыши, горячая клавиша, клавиша «Enter»).

Обозначим $E = \{e_i\}$, $i=1, n$ – упорядоченное по времени регистрации множество всех событий. Особо отметим, что каждое событие обладает непустым уникальным набором атрибутов, а, следовательно, является уникальным.

A. Классификация событий

Во множестве без повторений невозможно существование регулярно повторяющихся шаблонов как упорядоченных подмножеств. Поэтому, требуется предварительная классификация всех событий.

Обозначим $C = \{c_i\}$, $i=1, k$ – множество определенных классов событий. Классом событий будем считать произвольную совокупность событий, обладающих каким-либо определенным свойством или признаком. Например, возможна классификация событий: по виду, по названию исполненной команды (для командных событий), по области пользовательского интерфейса (в области верхнего меню, в основной рабочей области).

Для классификации событий необходимо многозначная классифицирующая функция f , которая может быть записана как: $f(e) \in C : f(e) = \{c \in C\}$.

Причина, почему функция является многозначной, исходит из многообразия возможных направлений исследований конкретного набора данных активности пользователей. Правила классификации и набор классов определяются экспертом и, в зависимости от целей исследования, часть событий может игнорироваться. Например, эксперта могут интересовать лишь командные события, т.е. связанные с вызовом какой-либо команды. В этом случае будем говорить, что произведена фильтрация событий прочих видов. С другой стороны, возможно сопоставление нескольких классов событий одному событию, в случае особого набора атрибутов.

В дальнейшем, под событиями будем понимать события именно после классификации, т.к. поиск последовательных шаблонов невозможен до этих действий по причине уникальности каждого события.

В. Шаблоны событий

Теория последовательных шаблонов подразумевает наличие множества кандидатов, для которых проверяется уровень поддержки. Определим понятия шаблона и поддержки шаблона для множества событий.

Обозначим $P = \{p_i\}$, $i = 1, r$ – множество кандидатов последовательных шаблонов событий. Шаблон представляет собой размещение элементов множества C с повторениями: $p_i = \{c_{i1} \dots c_{ij}\}$, где $j = 1, q_i$ – мощность размещения p_i , т.е. кол-во событий в шаблоне. Будем считать длиной шаблона количество событий в указанном размещении. Будем считать, что шаблон p входит в множество событий E , если все элементы p содержатся в E , при этом порядок элементов в подмножестве из E , соответствует порядку элементов p .

Обозначим μ_p – количество вхождений шаблона p в множество E , рассчитываемое как количество непересекающихся упорядоченных вхождений. Например, пусть имеется множество событий $\langle 2, 4, 2, 4, 2, 1, 2 \rangle$ и шаблон $\langle 2, 4, 2 \rangle$. Количество вхождений будет равно 1, т.к. 4-й элемент уже участвовал при подсчете.

Шаблон p называется поддерживаемым, если количество вхождений μ_p больше нуля.

Обозначим λ_p – поддержку шаблона p , рассчитываемую как: $\lambda_p = (\mu_p * q) / n$, где μ_p – количество вхождений шаблона, q – длина шаблона p , n – количество событий.

Поддержка принимает значение на отрезке $[0, 1]$. Таким образом, можно описать значение поддержки как долю содержания шаблона в событиях. Это необходимо для сравнения разных шаблонов по степени влияния на процесс взаимодействия пользователя с программным обеспечением, что невозможно сделать на основе лишь значения количества вхождений.

Например, пусть имеется размещение событий: $\langle 1, 2, 3, 2, 1, 1, 2, 3, 2, 1 \rangle$.

Рассчитаем значения μ и λ для следующих шаблонов:

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

Количество вхождений всех шаблонов одинаково и равно 2. Однако, поддержка шаблона p_4 равна 1, т.е. он полностью составляет всё множество событий, а значит является более вероятным кандидатом для пристального внимания эксперта. Повышение эффективности взаимодействия пользователя с программным интерфейсом в затрагиваемого подобным шаблоном кардинально может повысить эффективность всего процесса работы с программным обеспечением в целом.

С. Анализ шаблонов

Область интеллектуального анализа данных (англ. Data Mining) имеет хорошо развитую теория поиска ассоциативных правил и последовательных шаблонов. Адаптация и применение существующих методов (таких как AprioriAll [8], AprioriSome, DynamicSome и т.д.) позволит значительно оптимизировать расчет значений поддержки для сформированных кандидатов шаблонов.

Имея значения поддержки для каждого шаблона, эксперт может сконцентрироваться на наиболее значимых из них для процесса работы пользователей с программным обеспечением в целом. Набор шаблонов, при этом, будет зависеть от целей проводимого анализа.

Далее, эксперт может выдвинуть гипотезы о необходимых изменениях в пользовательском интерфейсе для повышения эффективности взаимодействия пользователей с программным обеспечением. При принятии решений эксперту необходимо учитывать множество различных факторов: особенности программного обеспечения, психологические факторы использования программного обеспечения и особенности пользователей.

Изменение пользовательского интерфейса повлечет изменение состава событий и последовательных шаблонов, т.к. изменится последовательность действий необходимых для достижения пользователями поставленных целей.

После внесения изменений в программный интерфейс возможен повторный сбор и анализ данных активности пользователей, что может подтвердить либо опровергнуть выдвинутую ранее гипотезу.

III. ЗАКЛЮЧЕНИЕ

Обзор показывает, что существует довольно обширный пласт исследований, в области анализа шаблонов поведения пользователей. Данная тематика имеет большой потенциал, в виду очевидной экономии ресурсов и времени при автоматическом анализе данных активности пользователей с целью обнаружения аспектов

программного интерфейса, требующих детального исследования.

Задача поиска повторяющихся шаблонов событий в данных активности пользователей имеет много общего с задачей поиска последовательных шаблонов из области интеллектуального анализа данных (англ. Data Mining). Следовательно, возможна адаптация и применение существующей теории, что позволит использовать созданные ранее наработки в новой области.

СПИСОК ЛИТЕРАТУРЫ

- [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. – October 1991. – Vol. 9, No. 4, pp. 309-335.
- [2] Данилов Н.А., Шульга Т.Э. Метод построения тепловой карты на основе точечных данных об активности пользователя приложения // Журнал «Прикладная информатика». 2015. Т. 10. № 2 (56). С. 49-58.
- [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. Volume 20, pp. 31-43, ISSN 1319-1578.
- [6] Ting, L., 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, September, 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). 77–87 pp.