

## ОЦЕНКА СТЕПЕНИ УДОБСТВА ИСПОЛЬЗОВАНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ В ЛОГИКЕ ТАЙЛОВ

В.В. Девятков

deviatkov@bmstu.ru

Е.А. Типсин

e.tipsin@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

---

### Аннотация

Представлены принципы методологии формальной автоматизированной оценки степени удобства пользовательских интерфейсов. В качестве математического аппарата для решения задач этой формальной проверки выбрана логика тайлов. Приведен обзор работ рассматриваемой области, объяснены причины выбора аппарата логики тайлов и рассмотрены принципы проверки (доказательства) удобства использования пользовательских интерфейсов. Показаны отличия разработанной методологии от известных методологий и ее достоинства. Принципы методологии формальной автоматизированной оценки степени удобства в логике тайлов продемонстрированы на простом примере. Рассмотрены развития предложенной методологии и перспективы ее использования

### Ключевые слова

Логика тайлов, верификация свойств, пользовательский интерфейс, степень удобства использования, юзабилити, оценка степени удобства использования, агенты, автоматизация оценки, человеко-машинное взаимодействие

Поступила 09.04.2019

Принята 19.06.2019

© Автор(ы), 2020

---

*Работа выполнена при поддержке Минобрнауки России (исследовательский проект № 2.5048.2017/8.9 «Автоматизация синтеза гибридных мультиагентных систем на основе формализации процессов спецификации и верификации»)*

**Введение.** В стандарте ISO/IEC 25010 введен термин «юзабилити» (от англ. *usability*), означающий «удобство и простоту использования, степень удобства использования», а также пригодность использования, эргономичность (способность приложения быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях для достижения установленных целей с необходимой результативностью, эффективностью и удовлетворенностью). Далее вместо термина «юзабилити» будем использовать словосочетание «степень удобства использования».

Объект исследования настоящей работы — степень удобства использования пользовательских интерфейсов, обеспечивающих взаимодей-

ствие пользователя с программными приложениями и осуществляемых в среде одной или совокупности веб-страниц посредством веб-браузера.

Существует много традиционных средств оценки степени удобства использования, среди которых проверка по контрольному списку [1], эвристическая оценка [2], мысленный прогон [3], множественный прогон [4], оценки функциональности [5], анализ задач [6], фокус группы [7], обзорные исследования и интервью, вопросники и анкетирование, фиксация зрительной активности пользователей [8] и др. Однако все эти оценки степени удобства использования довольно трудоемки, что проявляется в затратах времени на подготовку к сравнительному анализу, отбор респондентов и дальнейший анализ полученных результатов специалистами человеко-машинного взаимодействия, в некоторых случаях необходимо наличие специального оборудования. Чтобы снизить сложность и объем работ, выполняемых вручную, актуально направление автоматизации оценки степени удобства использования. В этом направлении существует три основных подхода к автоматизации оценки степени удобства использования визуальных пользовательских интерфейсов.

*Подход 1. Анализ свойств пользовательских логов.* Анализ осуществляется в хронологическом порядке по записи действий пользователя в процессе взаимодействия с реальными приложениями или их динамическими прототипами. Как правило, такими действиями являются перемещение курсора мыши, время, затраченное пользователем для взаимодействия с одной и той же страницей, переходы между страницами и пр.

Наиболее популярными сервисами, предоставляющими услуги по сбору пользовательских логов и их анализу, являются сервисы *Google Analytics* и *Yandex Metrics*. Эти сервисы обрабатывают собранные данные и на основе их могут визуализировать данные в виде:

- карты кликов, отображающей клики по всем элементам страниц приложения, тем самым помогая выявлять элементы, к которым применено наибольшее число кликов (максимально кликабельные), определять, какие элементы принимаются пользователем за ссылку, но таковой не являются. По картам кликов могут быть сформированы «тепловые карты», на которых места, соответствующие максимально кликабельным элементам, отображаются теплыми цветами, наименее кликабельным — холодными, а все другие — промежуточными цветами, зависящими от соответствующего им числа кликов;

- карты ссылок, отображающие число переходов по ссылке, долю переходов относительно других ссылок на странице;

– карты скроллинга, отображающие время внимания посетителей определенным элементам страницы сайта; позволяют эксперту человеко-машинного взаимодействия подбирать оптимальную длину страницы и правильно размещать важную информацию;

– карты путей по приложению, отображающие перемещение пользователей по страницам приложения; по этим картам эксперт человеко-машинного взаимодействия анализирует особенности последовательностей перемещения пользователей по страницам приложения.

Сервисы логирования представляют собой инструменты для сбора и визуализации данных для дальнейшего ручного анализа человеком, который чаще всего не является экспертом человеко-машинного взаимодействия и не всегда может правильно интерпретировать полученные данные. Анализ человеком этих данных является менее трудоемким, чем анализ аудиовизуальной информации, получаемой при проведении традиционных исследований степени удобства использования. Для того чтобы собирать данные о взаимодействии пользователя с приложением, оно должно быть запущено в рабочую эксплуатацию, что при его низком качестве влечет за собой потерю потенциальных клиентов. Трудности со сбором данных испытывают приложения с низкой посещаемостью, что приводит к большим срокам проведения оценки степени удобства использования. При использовании логирования практически невозможно точно выяснить причины ухода пользователя с приложения. Уход пользователя может быть как следствием того, что он достиг своей цели в результате работы с приложением, так и, наоборот, он не смог достичь цели.

Один из главных недостатков указанных сервисов логирования — недостаточная автоматизация процесса анализа логов. Существуют работы, в которых степень автоматизации логирования существенно выше. Вот некоторые из наиболее известных.

Метод анализа, который позволяет обнаружить в логах различных пользователей одинаковые последовательности действий, характерные для большинства из них, представлен в работе [9]. Эти одинаковые для большинства пользователей последовательности рассматриваются как шаблоны, из которых складывается взаимодействие каждого пользователя с приложением. Затем, используя критерий скорости достижения пользователем цели и генетический алгоритм, выявляются эффективные и неэффективные шаблоны взаимодействия пользователя с приложением и на основе этого делается вывод о степени удобства использования приложения (чем больше число использованных эффективных шаблонов, тем выше степень удобства использования приложения).

Инструмент, который позволяет вручную задать наилучшую с точки зрения ее создателя последовательность действий пользователя в процессе взаимодействия с приложением (эталонный лог), рассмотрен в работе [10]. После этого реальный лог, полученный в результате взаимодействия пользователя с приложением, сравнивается с эталонным логом по определенной мере. Чем ближе эталонный и реальный логи друг к другу, согласно этой мере, тем выше степень удобства использования приложения. Недостатком такого подхода является необъективность и интуитивность создания эталонного лога, а также увеличение временных затрат на его создание по мере повышения сложности приложения и числа его модификаций.

*Подход 2. Параметрический анализ свойств страниц.* Осуществляется проверка отдельных параметров страниц приложения на соответствие рекомендациям по дизайну, визуальному шуму и сложности интерфейса, а также выявление областей страницы, на которые пользователь прежде всего обратит внимание при просмотре страницы.

Оценивать страницы интерфейса на основе параметров графики и текста предложено в работе [11]. Здесь утверждается, что соотношение графики и текста играет важную роль в улучшении внешнего вида страниц, что, в свою очередь, оказывает влияние на эстетические чувства пользователей, которые они испытывают при работе с приложением. Для эстетического анализа в работе [2] построена нейронная сеть, анализирующая, как конкретная комбинация элементов пользовательского интерфейса веб-страницы влияет на эстетические чувства пользователей при работе с ней. Результатом применения нейронной сети является оценка страницы по пятибалльной шкале (нейросеть имеет пять выходов, каждому из которых соответствует одна из пяти взаимоисключающих оценок), которая помогает веб-дизайнерам оценить степень удобства использования приложения. Для построения нейронной сети применялась обучающая выборка, содержащая 96 страниц с их эстетическими оценками. Эстетические оценки определялись субъективно экспертами человеко-машинного взаимодействия с различным опытом работы. Задачи, для решения которых предназначались страницы, не учитывались. Для построения нейронной сети использовались только статические страницы.

Метод определения значимости элементов на страницах по движению глаз пользователя на различных страницах приложения предложен в работе [12]. Полученная информация может дать представление о том, как воспринимается интерфейс страницы и где находится самый важный или релевантный контент. Инструмент позволяет предсказывать важ-

ность различных элементов интерфейса, но без учета различных категорий пользователей и контекстов.

*Подход 3. Моделе-ориентированное доказательство наличия свойств взаимодействия.* Создается архитектура системы взаимодействия и формулируются свойства, которым эта архитектура должна удовлетворять. Затем эти свойства проверяются применительно к созданной архитектуре. Так, в работе [13] рассмотрены свойства мультимодальных интеллектуальных интерфейсов, реализация которых обеспечивает естественное «безбарьерное» взаимодействие пользователя с приложением. Если какое-либо свойство взаимодействия должно быть реализовано, то, во-первых, необходима его точная спецификация и, во-вторых, возможность формальной проверки этой спецификации применительно к конкретному взаимодействию. Кроме того, формулировка этих свойств должна носить объективный характер, признанный мировым сообществом.

Из известных реализаций моделе-ориентированного подхода доказательства наличия свойств можно также назвать работу [14], в которой представлен инструмент *МЕМО*, содержащий модель приложения и модель пользователя. Модель пользователя правильно взаимодействует с моделью приложения, если правила генерации ошибок, которые экспертно задает разработчик, не возникают в процессе выполнения задачи (в процессе взаимодействия). Другими словами, модель приложения не переходит в состояние, которое нежелательно для идеального взаимодействия. Задаваемые правила генерации ошибок являются субъективными правилами эксперта. Чем масштабнее интерфейс приложения (модель приложения), тем больше требуется времени, затрачиваемого на определение правил генерации ошибок. При внесении изменений в интерфейс приложения также необходимо редактировать правила генерации ошибок, добавляя новые и удаляя неактуальные.

В первом столбце таблицы перечислены, согласно приведенному аналитическому обзору [9], критерии для сравнения качества различных подходов оценки степени удобства использования приложений. В остальных столбцах для каждого из трех подходов знак «+» указывает на то, что подход удовлетворяет критерию, а знак «-» — не удовлетворяет. Согласно данным, приведенным в таблице, наибольшему числу критериев из рассмотренных подходов удовлетворяет моделе-ориентированное доказательство наличия свойств. Остаются неудовлетворенными три критерия: 1) возможность объяснения поведения пользователей; 2) отсутствие необходимости приглашения экспертов человекомашинного взаимодействия; 3) отсутствие субъективного фактора.

Даже с учетом неудовлетворенности перечисленных трех критериев подход 3 является наилучшим. Этот подход использует модели архитектуры и поведения системы, в которой реализуется взаимодействие. Если создатель какой-либо модели полагает, что ее архитектура и поведение обладает всеми необходимыми объективными свойствами, обеспечивающими необходимую степень удобства использования, то остается только доказать, что архитектура и поведение удовлетворяют этим свойствам.

**Сравнение подходов к автоматизации оценки степени удобства использования**

Критерий	Подход		
	1	2	3
Возможность проведения на ранней стадии разработки	–	–	+
Возможность объяснения поведения пользователей	–	–	–
Простота оценки после внесения изменений в интерфейс	–	+	–
Отсутствие необходимости приглашения экспертов человеко-машинного взаимодействия	–	–	–
Небольшие сроки проведения	–	+	+
Учет контекста использования приложения	–	–	+
Учет пользовательского опыта	–	–	+
Анализ динамических страниц	+	–	+
Отсутствие субъективного фактора	–	–	–

Авторы настоящей работы полагают, что устранение указанных недостатков неудовлетворенности критериев (возможность объяснения поведения пользователей, отсутствие необходимости приглашения экспертов человеко-машинного взаимодействия, отсутствие субъективного фактора) возможно на пути автоматизированного модели-ориентированного доказательства (вывода) наличия объективных свойств архитектуры и поведения взаимодействующих моделей пользователя и приложения в рамках адекватного исчисления, включающего в себя:

- формальный язык адекватного описания шаблонов поведения пользователя и исполнителя в процессе взаимодействия;
- формальное описание на этом языке объективных свойств архитектуры и поведения взаимодействующих моделей, гарантирующих высокую степень удобства использования приложений;
- множества состоятельных правил вывода в этом языке;
- стратегию вывода, гарантирующую полноту доказательства наличия свойств архитектуры и поведения взаимодействующих моделей.

Формальное описание объективных свойств архитектуры и поведения взаимодействующих моделей, гарантирующих высокую степень удобства использования приложений, позволит избавиться от необходимости приглашения экспертов человекомашинного взаимодействия и, как следствие, от их субъективного мнения.

Стратегия вывода, гарантирующая полноту доказательства наличия свойств архитектуры и поведения взаимодействующих моделей на основе состоятельных правил вывода, позволит выявлять степень удобства использования и причины неудобства.

Автоматизация стратегии вывода обеспечит простоту оценки степени удобства приложения после внесения изменений в процесс взаимодействия.

По мнению авторов работы, наиболее адекватным для автоматизации оценки степени удобства использования приложений является язык логики тайлов. Это мнение основано, в частности, на том, что тайл, как конструкция, позволяет, с одной стороны, естественным образом отображать элементы страниц, а с другой, — интерфейсы между элементами, осуществляемыми в результате определенных действий пользователя. Кроме того, из тайлов по определенным правилам могут компоноваться допустимые последовательности взаимодействия пользователя с системой, которые служат средством оценки степени удобства использования.

Принципы доказательства каких-либо свойств систем на основе логики тайлов известны достаточно давно [15]. Для того чтобы создать язык логики тайлов для рассматриваемого случая, необходимо, прежде всего, создать базовое множество тайлов, позволяющих представлять поведение приложения (агента-исполнителя) и респондента (агента-пользователя).

Далее приведены основные понятия, связанные с тайлами, затем представлена суть формального описания моделей приложения (агента-исполнителя) и респондента (агента-пользователя) с помощью тайлов. Введены критерии, по которым будет оцениваться степень удобства приложения. Приведен простой пример выявления степени удобства приложения в простом языке логики тайлов, а также направления дальнейших работ в области автоматизации оценки удобства приложений на основе логики тайлов.

**Тайлы и операции над ними.** Достоинства логики тайлов связаны со структурой ее правил вывода, каждое из которых может быть представлено в виде тайла (*tile*) (рис. 1)

В символическом виде тайл может быть представлен следующим образом:

$$\alpha : ?i_i, ?s_i, !o_i \xrightarrow{?a_i} ?i_o, !s_o, !o_o. \quad (1)$$

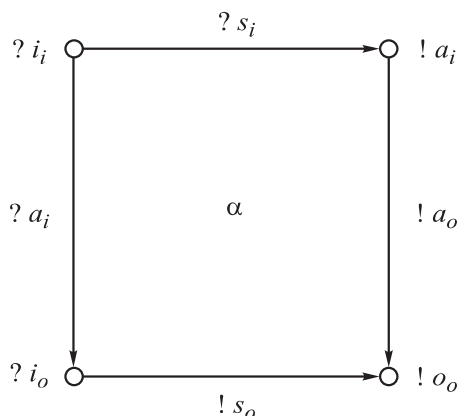


Рис. 1. Структура тайла:

$\alpha$  — имя тайла;  $?s_i$  — начальная конфигурация тайла;  $!s_o$  — конечная конфигурация тайла;  $?a_i$  — восприятие тайла;  $!a_o$  — реакция тайла;  $?i_i$  — начальный входной интерфейс тайла;  $!o_i$  — начальный выходной интерфейс тайла;  $?i_o$  — конечный входной интерфейс тайла;  $!o_o$  — конечный выходной интерфейс тайла

Каждый тайл описывает отдельный элемент поведения системы в терминах возможных взаимодействий с внешней средой. В целом поведение системы выглядит как скоординированное взаимодействие отдельных тайлов. Тайлы могут композироваться горизонтально, вертикально и параллельно для того, чтобы задавать более сложное поведение систем. Та или иная логика тайлов получается в результате введения множества базовых тайлов и некоторой совокупности вспомогательных тайлов, над которыми могут совершаться операции композиции.

**Модель оценки степени удобства.** Для демонстрации принципов методологии автоматизации процесса оценки степени удобства пользовательских интерфейсов использована архитектура, моделирующая пользовательский интерфейс и состоящая из агента-исполнителя и агента-пользователя. Агент-исполнитель имеет два канала: 1) входной канал  $UE$ , являющийся выходным для агента-пользователя; 2) выходной канал  $EU$ , являющийся входным для агента-пользователя (рис. 2). Во входной канал агента-исполнителя агент-пользователь может помещать сообщение, которое может забирать агент-исполнитель. В выходной канал агент-исполнитель может помещать сообщение для агента-пользователя. Здесь и далее агента-исполнителя и агента-пользователя будем называть исполнителем и пользователем соответственно.





Рис. 2. Каналы взаимодействия пользователя и исполнителя

Пара  $[UE(m_u), EU(m_e)]$ , где  $UE(m_u)$  означает, что в канале  $UE$  находится сообщение  $m_u$ , а в канале  $EU$  — сообщение  $m_e$ , является в рассматриваемом случае конфигурацией, упоминаемой при определении тайла (см. рис. 1). Если безразлично, какое сообщение находится в том или ином канале, то вместо указания конкурентного сообщения будем ставить знак «\_». Если в канале ничего нет, то будем использовать знак « $\emptyset$ ».

Взаимодействие исполнителя и пользователя начинается после получения от пользователя уведомления о начале взаимодействия. Если исполнитель дал согласие на взаимодействие, то оно продолжается. В противном случае взаимодействие прекращается. Пользователь прекращает взаимодействие путем отправки соответствующего сообщения исполнителю.

**Архитектура пользовательских интерфейсов.** Как уже было отмечено выше, пользовательский интерфейс — процесс, обеспечивающий взаимодействие пользователя с исполнителем (программным приложением). Для того чтобы формализовать это взаимодействие, введем следующие понятия и обозначения.

Архитектурой пользовательского интерфейса назовем пятерку  $I = \{P, B, E, T_E, T_U\}$ , где  $P$  — множество страниц  $p$  интерфейса,  $p \in P$ ;  $B(p)$  — множество блоков  $b$  страницы  $p$ ,  $b \in B(p)$ ;  $E(b)$  — множество элементов  $e$  блока  $b$ ,  $e \in E(b)$ ;  $T_E, T_U$  — множества тайлов, описывающих поведение исполнителя и пользователя. Страницы, блоки и элементы имеют уникальное обозначение  $p, b, e$  с числовыми индексами, играющее роль объектной (индивидуальной) переменной, которую будем называть просто объектом. Значение объектной переменной — объектная константа, являющаяся набором значений таких атрибутов, как пояснение, поле ввода, поле вывода, кнопка, изображение, список, видеоматериал, аудиоматериал, всплывающая подсказка, форма, цвет, яркость и др. Каждый атрибут может иметь достаточно сложную структуру. Как правило, многие атрибуты в их явном виде скрыты от пользователя.

Так, на рис. 3 показан простой пример страницы  $p_1$  пользовательского интерфейса, содержащий блок  $b_1 =$  Преобразование систем счисления (ПОС) и элементы  $e_1 =$  Исходная система счисления (ИСС),  $e_2 =$  Исходное число (ИЧ),  $e_3 =$  Результирующая система счисления (РСС),  $e_4 =$  Резуль-

тирующее число (РЧ). Здесь текст «Преобразование систем счисления» поясняет назначение блока  $b_1$ , тексты «Исходная систем счисления», «Исходное число» являются полями ввода элементов  $e_1, e_2$ , а тексты «Результирующая система счисления», «Результирующее число» — полями вывода элементов  $e_3, e_4$ .

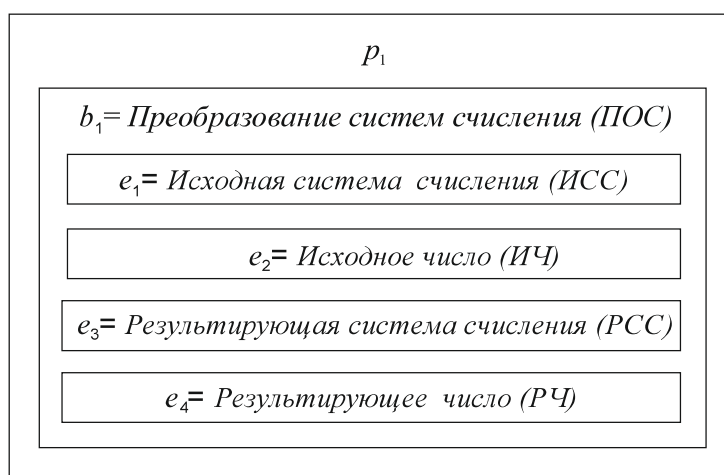


Рис. 3. Пример страницы пользовательского интерфейса

**Тайлы для описания поведения исполнителя.** Поведение исполнителя описывается двумя типами тайлов.

Тайл «Указание исполнителем объекта (УИО)», находясь в конфигурации  $[UE(m_u), EU(-)]$ , осуществляет восприятие сообщения  $m_u$ , именуемое тот или иной объект, к которому пользователю необходим доступ, выдает реакцию  $!m_e$ , указывающую на объект  $!m_e$ , к которому пользователю разрешается доступ, устанавливает конечный выходной интерфейс  $!o_o = m_e$ . Конечная конфигурация тайла «Указание исполнителем объекта» при этом становится конфигурацией  $[UE(m_u), EU(m_e)]$ :

$$\begin{aligned} \text{УИО} : ?i_i = m_u, ?s_i = [UE(m_u), EU(-)], ?o_i = - \frac{?m_u}{!m_e} ?i_o = m_u, \\ !s_o = [UE(m_u), EU(m_e)], !o_o = m_e. \end{aligned} \quad (2)$$

Тайл «Обработка объекта исполнителем (ООИ)» всегда выполняется после тайла «Указание исполнителем объекта», согласно горизонтальной композиции тайлов, и позволяет помещать реакцию исполнителя (сообщение исполнителя)  $m_e^*$  в выходной канал  $EU$  в ответ на восприятие собственной реакции  $m_e$  и в ответ на восприятие сообщения пользова-

теля  $m_u$  тайлом «Указание исполнителем объекта», а также при условии, что его начальный и конечный входные интерфейсы совпадают с начальным и конечным выходными интерфейсами тайла «Указание исполнителем объекта». Начальная конфигурация  $[UE(m_u), EU(m_e)]$  тайла «Обработка объекта исполнителем» переходит в конечную конфигурацию  $[UE(m_u), EU(m_e^*)]$ :

$$\begin{aligned} \text{ООИ} : ?i_i = m_u, ?s_i = [UE(m_u), EU(m_e)], !o_i = m_e \frac{?m_e}{!m_e^*} ?i_o = m_u, \\ !s_o = [UE(m_u), EU(m_e^*)], !o_o = m_e^*. \end{aligned} \quad (3)$$

**Тайлы для описания поведения пользователя.** Поведение исполнителя описывается тремя типами тайлов.

Тайл «Инициализация пользователем взаимодействия (ИПВ)» позволяет начать взаимодействие пользователя и исполнителя. Согласно этому тайлу, его начальной конфигурацией является  $[UE(\emptyset), EU(\emptyset)]$  (каналы пустые), его начальные входной и выходной интерфейсы и восприятие также являются пустыми. Тайл позволяет выдавать реакцию  $!m_u$ , именуемую объект, с которого следует начинать взаимодействие. В результате конечная конфигурация тайла становится равной  $[UE(m_u), EU(\emptyset)]$ , а конечный выходной интерфейс  $o_o = m_u$ , означающий разрешение доступа к объекту  $m_u$ :

$$\begin{aligned} \text{ИПВ} : ?i_i = \emptyset, ?s_i = [UE(\emptyset), EU(\emptyset)], !o_i = \emptyset \frac{? \emptyset}{!m_u} ?i_o = \emptyset, \\ !s_o = [UE(m_u), EU(\emptyset)], !o_o = m_u. \end{aligned} \quad (4)$$

Тайл «Указание пользователем объекта» позволяет формировать выходной конечный интерфейс пользователя в ответ на восприятие пользователя (сообщение исполнителя). Тайл, находясь в конфигурации  $[UE(m_u), EU(m_e)]$ , осуществляет восприятие сообщения  $m_u^*$ , именуемое тот или иной объект, к которому разрешен доступ исполнителю с помощью тайла «Указание исполнителем объекта», начальный выходной интерфейс  $?o_i = m_e^*$  выдает реакцию  $!m_e$ , запрашивающую доступ к объекту  $m_e$ . Конечная конфигурация тайла «Указание исполнителем объекта» при этом становится конфигурацией  $[UE(m_u^*), EU(m_e)]$ :

$$\begin{aligned} \text{УПО} : ? i_i = m_u, ? s_i = [UE(m_u), EU(m_e)], ! o_i = m_e \frac{? m_u^*}{! m_e} ? i_o = m_u^*, \\ ! s_o = [UE(m_u^*), EU(m_e)], ! o_o = m_e. \end{aligned} \quad (5)$$

Тайл «Обработка объекта пользователем» выполняется после тайла «Указание пользователем объекта», согласно вертикальной композиции тайлов, и выдает реакцию пользователя  $m'_u$  (результат обработки объекта  $m_u^*$ ) в выходной канал  $UE$  в ответ на восприятие собственной реакции  $! m_u^*$ , подтверждающей восприятие сообщения  $m_u^*$  пользователя тайлом «Указание исполнителем объекта» и при условии, что его начальные входной и выходной интерфейсы совпадают с конечными входным и выходным интерфейсами тайла «Указание пользователем объекта». Начальная конфигурация  $[UE(m_u^*), EU(m_e)]$  тайла «Обработка объекта пользователем» переходит в конечную конфигурацию  $[UE(m'_u), EU(m_e)]$ :

$$\begin{aligned} \text{ООП} : ? i_i = m_u^*, ? s_i = [UE(m_u^*), EU(m_e)], ! o_i = m_e \frac{? m_u^*}{! m'_u} ? i_o = m'_u, \\ ! s_o = [UE(m'_u), EU(m_e)], ! o_o = m_e. \end{aligned} \quad (6)$$

**Пример взаимодействия пользователя и исполнителя в логике тайлов.** Для страницы, пример которой приведен на рис. 3, введем следующие тайлы.

Тайл «Инициализация пользователем взаимодействия», задающий конечный выходной интерфейс  $o_o = p_1$ , т. е. страницу  $p_1$ , с которой должно начаться взаимодействие:

$$\begin{aligned} \text{ИПВ} : ? i_i = \emptyset, ? s_i = [UE(\emptyset), EU(\emptyset)], ! o_i = \emptyset \frac{? \emptyset}{! p_1} ? i_o = p_1, \\ ! s_o = [UE(p_1), EU(\emptyset)], ! o_o = p_1. \end{aligned} \quad (7)$$

Тайл «Указание исполнителем объекта», подтверждающий своей реакцией  $! p_1$  согласие исполнителя на обработку пользователем страницы  $p_1$ :

$$\begin{aligned} \text{УИО} : ? i_i = \emptyset, ? s_i = [UE(p_1), EU(\emptyset)], ! o_i = p_1 \frac{? p_1}{! p_1} ? i_o = p_1, \\ ! s_o = [UE(p_1), EU(p_1)], ! o_o = p_1. \end{aligned} \quad (8)$$

Тайл «Указание пользователем объекта», запрашивающий своей реакцией  $! b_1$  доступ к блоку  $b_1$ :

$$\begin{aligned} \text{УПО} : ?i_i = p_1, ?s_i = [UE(p_1), EU(p_1)], !o_i = p_1 \frac{?p_1}{!b_1} ?i_o = p_1, \\ !s_o = [UE(b_1), EU(p_1)], !o_o = b_1. \end{aligned} \quad (9)$$

Тайл «Указание исполнителем объекта», подтверждающий своей реакцией  $!b_1$  согласие исполнителя на обработку пользователем блока  $b_1$ :

$$\begin{aligned} \text{УИО} : ?i_i = p_1, ?s_i = [UE(b_1), EU(p_1)], !o_i = b_1 \frac{?b_1}{!b_1} ?i_o = b_1, \\ !s_o = [UE(b_1), EU(b_1)], !o_o = b_1. \end{aligned} \quad (10)$$

Тайл «Указание пользователем объекта», запрашивающий своей реакцией  $!e_1$  доступ к элементу  $e_1$ :

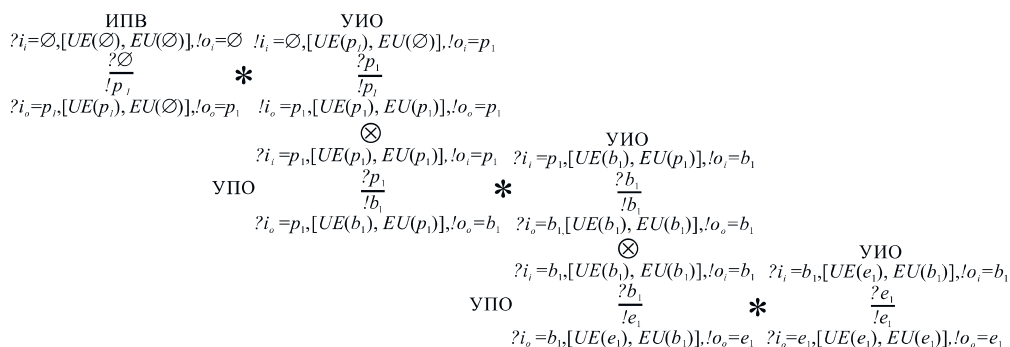
$$\begin{aligned} \text{УПО} : ?i_i = b_1, ?s_i = [UE(b_1), EU(b_1)], !o_i = b_1 \frac{?b_1}{!e_1} ?i_o = b_1, \\ !s_o = [UE(b_1), EU(e_1)], !o_o = e_1. \end{aligned} \quad (11)$$

Тайл «Указание исполнителем объекта», подтверждающий своей реакцией  $!e_1$  согласие исполнителя на обработку пользователем элемента  $e_1$ :

$$\begin{aligned} \text{УИО} : ?i_i = b_1, ?s_i = [UE(b_1), EU(e_1)], !o_i = b_1 \frac{?e_1}{!e_1} ?i_o = e_1, \\ !s_o = [UE(e_1), EU(e_1)], !o_o = e_1. \end{aligned} \quad (12)$$

Тайлов (7)–(12) достаточно для моделирования поведения пользователя и исполнителя в процессе навигации от страницы  $p_1$  к элементу  $e_1$ , так как на этом этапе никакой обработки объектов не осуществляется. Это поведение представлено композицией тайлов (рис. 4). Для компактного представления композиции тайлы представлены несколько в ином виде, а именно в виде трех строк, где первая строка содержит начальный входной и выходной интерфейсы и начальную конфигурацию, вторая строка — восприятие и реакцию, а третья строка — конечный входной и выходной интерфейсы и конечную конфигурацию.

Для дальнейшего моделирования взаимодействия пользователя и исполнителя требуется введение тайлов обработки объектов. Для всех элементов, показанных на рис. 3, схема обработки идентична и заключается в следующем. После указания исполнителем пользователю с помощью тайла «Установка исполнителем объектов» элемента, к которому разрешен доступ, исполнитель с помощью тайла «Обработка объекта исполнителем» указывает поле для ввода информации пользователем. Пользо-



**Рис. 4.** Композиция тайлов, моделирующая поведение пользователя и исполнителя при навигации от страницы  $p_1$  к элементу  $e_1$ :

символом «\*» обозначена горизонтальная композиция, а символом «⊗» — вертикальная

атель с помощью тайла «*Обработка объекта пользователем*» вводит в это поле информацию. Исполнитель после восприятия этой информации указывает пользователю на следующий элемент, к восприятию которого пользователь может перейти. Введем требуемые тайлы только для элемента  $e_1$  и покажем, как они компонируются для моделирования поведения пользователя и исполнителя.

Тайл «*Обработка объекта исполнителем*» выполняется после тайла (12), согласно горизонтальной композиции тайлов, и позволяет помещать реакцию исполнителя *ИСС* в выходной канал *EU* в ответ на восприятие собственной реакции  $e_1$  тайла (12) и при условии, что его начальный и конечный входные интерфейсы совпадают с начальным и конечным выходными интерфейсами тайла «*Обработка объекта исполнителем*». Начальная конфигурация  $[UE(e_1), EU(e_1)]$  тайла «*Обработка объекта исполнителем*» переходит в конечную конфигурацию  $[UE(e_1), EU(ИСС)]$ :

$$OOI : ?i_i = b_1, ?s_i = [UE(e_1), EU(e_1)], !o_i = e_1 \frac{?e_1}{!ИСС} ?i_o = e_1, \quad (13)$$

$$!s_o = [UE(e_1), EU(ИСС)], !o_o = ИСС.$$

Тайл «*Обработка объекта пользователем*» выполняется после тайла «*Обработка объекта исполнителем*», согласно вертикальной композиции тайлов, и формирует реакцию пользователя в выходной канал *UE* в ответ на восприятие реакции *ИСС* тайла (13), указывающую значение поля *ИСС* (объекта *ИСС*). Например, этой реакцией может быть *Двоичная система счисления (ДСС)*. Начальная конфигурация  $[UE(e_1), EU(ИСС)]$  тайла «*Обработка объекта пользователем*» переходит в конечную конфигурацию  $[UE(ДСС), EU(ИСС)]$ :

$$\begin{aligned} \text{ООП} : ?i_i = e_1, ?s_i = [UE(e_1), EU(ИСС)], !o_i = ИСС \frac{?ИСС}{!ДСС} ?i_o = ДСС, \\ !s_o = [UE(ДСС), EU(ИСС)], !o_o = ИСС. \end{aligned} \quad (14)$$

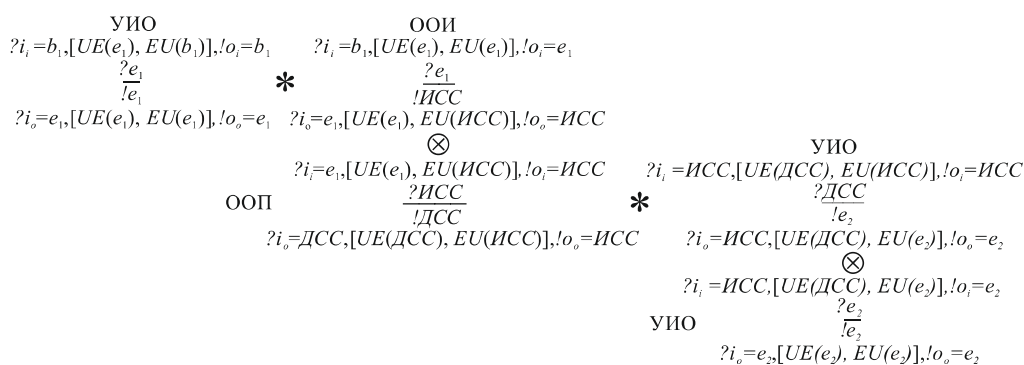
Тайл «Указание исполнителем объекта» своей реакцией  $e_2$  указывает, что следующим элементом, подлежащим обработке пользователем, является элемент  $e_2$ :

$$\begin{aligned} \text{УИО} : ?i_i = ИСС, ?s_i = [UE(ДСС), EU(ИСС)], !o_i = \\ = ИСС \frac{?ДСС}{!e_2} ?i_o = ИСС, \\ !s_o = [UE(ДСС), EU(e_2)], !o_o = e_2. \end{aligned} \quad (15)$$

Тайлом «Указание пользователем объекта» пользователь подтверждает переход к обработке элемента  $e_2$ :

$$\begin{aligned} \text{УИО} : ?i_i = ИСС, ?s_i = [UE(ДСС), EU(e_2)], !o_i = e_2 \frac{?e_2}{!e_2} ?i_o = e_2, \\ !s_o = [UE(e_2), EU(e_2)], !o_o = e_2. \end{aligned} \quad (16)$$

Композиция тайлов (13)–(16) показана на рис. 5 и демонстрирует поведение пользователя и исполнителя, в результате которого осуществляется переход от элемента  $e_1$  к элементу  $e_2$ , обработка которого аналогична обра-



**Рис. 5.** Композиция тайлов, моделирующая поведение пользователя и исполнителя при обработке элемента  $e_1$

ботке элемента  $e_1$ . Переходы к другим элементам рассматриваемого примера и их обработке аналогичны продемонстрированным. Для компактной композиции тайлы представлены в несколько другом виде (см. рис. 4).

**Критерии оценки степени удобства.** Критерии оценки степени удобства пользовательских интерфейсов могут быть качественными и количественными и позволяют делать вывод, насколько интерфейс удобен, а также сравнивать его с другими проектами пользовательских интерфейсов.

Простейшие критерии оценки степени удобства пользовательских интерфейсов — оценки достижения пользователями своих целей при работе с приложениями. Дж. Нильсен (J. Nielsen, [5]) предложил три такие оценки:

- 1) пользователь достиг своей цели;
- 2) пользователь столкнулся с проблемами, но достиг своей цели;
- 3) пользователь не достиг цели.

Цели в логике тайлов могут формулироваться как целевые тайлы, достижимость которых проверяется наличием композиции, в результате которой целевой тайл достигается. Если такой композиции не существует, то цель считается недостигнутой. В случае достижения цели простейшей оценкой степени удобства может быть, например, число тайлов, требуемых для достижения цели. Чем меньше тайлов для этого требуется, тем выше степень удобства использования. Более точные критерии оценки степени удобства использования требуют учета различных качественных и количественных свойств пользовательских интерфейсов, таких как сложность достижения конечной цели, сложность достижения промежуточных целей, выявление причин, приводящих к слишком высокой сложности достижения целей, причина достижения цели и т. п. Все это требует создания значительных библиотек тайлов, их классификации по группам пользователей, введения частных критериев оценки выполнения отдельных тайлов, по которым формируются общие оценки, и т. п.

Задача, которая поставлена в настоящей работе, не предполагает детального рассмотрения всех критериев оценки степени удобства использования пользовательских интерфейсов. Здесь рассмотрен только один критерий, который оценивает степень удобства использования приложения по результату выполнения задачи (выполнена — приложение имеет высокую степень удобства, в противном случае низкую).

**Заключение.** Приведен аналитический обзор подходов к автоматизации оценки степени удобства использования приложений, в рамках которого выявлены преимущества и недостатки каждого подхода. Предложен формальный модели-ориентированный метод оценки степени удобства использования, основанный на использовании логики тайлов и позволяющий устранить недостатки известных подходов. В логике тайлов создается модель взаимодействия пользователя и исполнителя (приложения). Мо-



дель композируется из тайлов в соответствии с определенными правилами, которые служат средством оценки удобства использования. Принципы методологии проиллюстрированы на примере одностраничного приложения. В дальнейшем будет создана полномасштабная модель взаимодействия в логике тайлов, учитывающая особенности поведения различных групп пользователей при взаимодействии с реальными приложениями, увеличено число критериев оценки степени удобства использования. Фактически, оценка степени удобства использования в логике тайлов — извлечение знаний о поведении взаимодействующих агентов, крайне востребованных как предмет аналитики на современном этапе развития интеллектуальных информационных систем. Изложены принципы извлечения знаний о поведении взаимодействующих агентов, основанные на использовании паттернов локального поведения агентов в логике тайлов, состав которых формирует общее поведение взаимодействующих агентов. Представление этих принципов осуществляется на уровне композиции тайлов, но без формального доказательства свойств реального поведения взаимодействующих агентов. Конкретизация такого доказательства свойств для случая выявления наличия определенных свойств интеллектуальных интерфейсов в языке логического программирования *PROLOG* дана в работе [13]. В основе процедур доказательства свойств взаимодействия агентов на языке *PROLOG* лежит механизм унификации, который может служить средством автоматизированной проверки свойств поведения агентов, представляемого различными композициями тайлов. Кроме того, по аналогии с той же работой интересующие свойства поведения взаимодействующих агентов могут быть формально выражены на модальных языках для последующей их формальной проверки.

## ЛИТЕРАТУРА

- [1] Reed P., Holdaway K., Isensee S., et al. User interface guidelines and standards: progress, issues, and prospects. *Interact. Comput.*, 1999, vol. 12, iss. 2, pp. 119–142. DOI: [https://doi.org/10.1016/S0953-5438\(99\)00008-9](https://doi.org/10.1016/S0953-5438(99)00008-9)
- [2] Nielsen J., Molich R. Heuristic evaluation of user interfaces. *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 1990, pp. 249–256. DOI: <https://doi.org/10.1145/97243.97281>
- [3] Rieman J., Franzke M., Redmiles D. Usability evaluation with the cognitive walkthrough. *Conf. Companion on Human Factors in Computing Systems*, 1995, pp. 387–388. DOI: <https://doi.org/10.1145/223355.223735>
- [4] Bias R.G. The pluralistic usability walkthrough: coordinated empathies. In: *Usability Inspection Methods*. Wiley & Sons, 1994, pp. 63–76.

- [5] Nielsen J. Usability inspection methods. *Conf. Companion on Human Factors in Computing Systems*, 1994, pp. 413–414. DOI: <https://doi.org/10.1145/259963.260531>
- [6] Hackos J.T., Redish J. User and task analysis for interface design. Wiley, 1998.
- [7] Nielsen J. The use and misuse of focus groups. *IEEE Softw.*, 1997, vol. 14, iss. 1, pp. 94–95. DOI: <https://doi.org/10.1109/52.566434>
- [8] Jacob R.J., Karn K.S. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In: *The Mind's Eye. Cognitive and Applied Aspects of Eye Movement Research*. Elsevier Science, 2003, pp. 573–605.
- [9] Liu Z., Wang Y., Dontcheva M., et al. Patterns and sequences: interactive exploration of clickstreams to understand common visitor paths. *IEEE Trans. Vis. Comput. Graphics*, 2017, vol. 23, no. 1, pp. 321–330. DOI: <https://doi.org/10.1109/TVCG.2016.2598797>
- [10] Baker S., Au F., Dobbie G., et al. Automated usability testing using HUI analyzer. *Proc. ASWEC*, 2008, pp. 579–588. DOI: <https://doi.org/10.1109/ASWEC.2008.4483248>
- [11] Lin Y.-C., Yeh C.-H., Wei C.-C. How will the use of graphics affect visual aesthetics? A user-centered approach for web page design. *Int. J. Hum. Comput. Stud.*, 2013, vol. 71, iss. 3, pp. 217–227. DOI: <https://doi.org/10.1016/j.ijhcs.2012.10.013>
- [12] Bylinskii Z., Kim N.W., O'Donovan P., et al. Learning visual importance for graphic designs and data visualizations. *Proc. 30th Ann. ACM Symp. User Interface Software and Technology*, 2017, pp. 57–69. DOI: <https://doi.org/10.1145/3126594.3126653>
- [13] Девятков В.В. Верификация свойств интеллектуальных интерфейсов в логике тайлов. *Вестник М ГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2016, № 3 (108), с. 65–87. DOI: <https://doi.org/10.18698/0236-3933-2016-3-65-87>
- [14] Engelbrecht K.P., Kruppa M., Möller S., et al. MeMo workbench for semi-automated usability testing. *9th Ann. Conf. Int. Speech Communication Association*, 2008, pp. 1662–1665.
- [15] Bruni R., Meseguer J., Montanari U. Process and term tile logic. Technical Report TR-98-09. Universiteta de Pisa, 1998.
- [16] Mahyavanshi N., Patil M., Kulkarni V. Enhancing web usability using user behavior and cognitive study. *Int. J. Comput. Appl.*, 2017, vol. 164, no. 2, pp. 27–31. DOI: <https://doi.org/10.5120/ijca2017913594>
- [17] Bradley S. Design layouts: Gutenberg diagram, Z-pattern, and F-pattern. *vanseodesign.com: веб-сайт*. URL: <https://vanseodesign.com/web-design/3-design-layouts/> (дата обращения: 18.09.2019).

**Девятков Владимир Валентинович** — д-р техн. наук, профессор, заведующий кафедрой «Информационные системы и телекоммуникации» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

**Типсин Евгений Андреевич** — аспирант кафедры «Информационные системы и телекоммуникации» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

**Просьба ссылаться на эту статью следующим образом:**

Девятков В.В., Типсин Е.А. Оценка степени удобства использования пользовательских интерфейсов в логике тайлов. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2020, № 1 (130), с. 64–84.

DOI: <https://doi.org/10.18698/0236-3933-2020-1-64-84>

**ASSESSMENT OF USER INTERFACE USABILITY  
IN TERMS OF TILE LOGIC**

V.V. Devyatkov

deviatkov@bmstu.ru

E.A. Tipsin

e.tipsin@mail.ru

**Bauman Moscow State Technical University, Moscow, Russian Federation**

**Abstract**

The paper presents methodological principles for a formal automated assessment of user interface usability. We chose tile logic as the mathematical tool for solving the problems pertaining to this formal check. The paper reviews previous publications in the field under consideration, explains the reasons behind selecting the method of tile logic, and considers the principles governing verification (proof) of user interface usability. We highlight the differences between the procedure developed and those known previously, and its advantages. We use a simple example to show the principles of formal automated assessment of user interface usability in terms of tile logic. We describe the development of the procedure proposed and the prospects of employing it

**Keywords**

*Tile logic, property verification, user interface, usability, usability assessment, agents, assessment automation, human-computer interaction*

Received 09.04.2019

Accepted 19.06.2019

© Author(s), 2020

*The study was supported by the Ministry of Education and Science of Russian Federation (research project no. 2.5048.2017/8.9, Automating the synthesis of multi-agent hybrid system by formalising specification and verification processes)*

**REFERENCES**

[1] Reed P., Holdaway K., Isensee S., et al. User interface guidelines and standards: progress, issues, and prospects. *Interact. Comput.*, 1999, vol. 12, iss. 2, pp. 119–142. DOI: [https://doi.org/10.1016/S0953-5438\(99\)00008-9](https://doi.org/10.1016/S0953-5438(99)00008-9)

- [2] Nielsen J., Molich R. Heuristic evaluation of user interfaces. *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 1990, pp. 249–256.  
DOI: <https://doi.org/10.1145/97243.97281>
- [3] Rieman J., Franzke M., Redmiles D. Usability evaluation with the cognitive walkthrough. *Conf. Companion on Human Factors in Computing Systems*, 1995, pp. 387–388. DOI: <https://doi.org/10.1145/223355.223735>
- [4] Bias R.G. The pluralistic usability walkthrough: coordinated empathies. In: *Usability Inspection Methods*. Wiley & Sons, 1994, pp. 63–76.
- [5] Nielsen J. Usability inspection methods. *Conf. Companion on Human Factors in Computing Systems*, 1994, pp. 413–414. DOI: <https://doi.org/10.1145/259963.260531>
- [6] Hackos J.T., Redish J. User and task analysis for interface design. Wiley, 1998.
- [7] Nielsen J. The use and misuse of focus groups. *IEEE Softw.*, 1997, vol. 14, iss. 1, pp. 94–95. DOI: <https://doi.org/10.1109/52.566434>
- [8] Jacob R.J., Karn K.S. Eye tracking in human-computer interaction and usability research: ready to deliver the promises. In: *The Mind's Eye. Cognitive and Applied Aspects of Eye Movement Research*. Elsevier Science, 2003, pp. 573–605.
- [9] Liu Z., Wang Y., Dontcheva M., et al. Patterns and sequences: interactive exploration of clickstreams to understand common visitor paths. *IEEE Trans. Vis. Comput. Graphics*, 2017, vol. 23, no. 1, pp. 321–330.  
DOI: <https://doi.org/10.1109/TVCG.2016.2598797>
- [10] Baker S., Au F., Dobbie G., et al. Automated usability testing using HUI analyzer. *Proc. ASWEC*, 2008, pp. 579–588.  
DOI: <https://doi.org/10.1109/ASWEC.2008.4483248>
- [11] Lin Y.-C., Yeh C.-H., Wei C.-C. How will the use of graphics affect visual aesthetics? A user-centered approach for web page design. *Int. J. Hum. Comput. Stud.*, 2013, vol. 71, iss. 3, pp. 217–227. DOI: <https://doi.org/10.1016/j.ijhcs.2012.10.013>
- [12] Bylinskii Z., Kim N.W., O'Donovan P., et al. Learning visual importance for graphic designs and data visualizations. *Proc. 30th Ann. ACM Symp. User Interface Software and Technology*, 2017, pp. 57–69.  
DOI: <https://doi.org/10.1145/3126594.3126653>
- [13] Devyatkov V.V. Verification of intelligent interface properties in the tiles logic. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2016, no. 3 (108), pp. 65–87 (in Russ.).  
DOI: <https://doi.org/10.18698/0236-3933-2016-3-65-87>
- [14] Engelbrecht K.P., Kruppa M., Möller S., et al. MeMo workbench for semi-automated usability testing. *9th Ann. Conf. Int. Speech Communication Association*, 2008, pp. 1662–1665.
- [15] Bruni R., Meseguer J., Montanari U. Process and term tile logic. Technical Report TR-98-09. Universiteta de Pisa, 1998.
- [16] Mahyavanshi N., Patil M., Kulkarni V. Enhancing web usability using user behavior and cognitive study. *Int. J. Comput. Appl.*, 2017, vol. 164, no. 2, pp. 27–31.  
DOI: <https://doi.org/10.5120/ijca2017913594>

[17] Bradley S. Design layouts: Gutenberg diagram, Z-pattern, and F-pattern. *vanseodesign.com: website*.

Available at: <https://vanseodesign.com/web-design/3-design-layouts/> (accessed: 18.09.2019).

**Devyatkov V.V.** — Dr. Sc. (Eng.), Professor, Head of Department of Information Systems and Telecommunications, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

**Tipsin E.A.** — Post-Graduate Student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

**Please cite this article in English as:**

Devyatkov V.V., Tipsin E.A. Assessment of user interface usability in terms of tile logic. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2020, no. 1 (130), pp. 64–84 (in Russ.).

DOI: <https://doi.org/10.18698/0236-3933-2020-1-64-84>