

Научная статья

УДК 004.5

DOI:10.31854/1813-324X-2023-9-6-102-115



Эффективная модель интерфейса взаимодействия пользователя с информационным сервисом запросного типа

Павел Андреевич Курта, expert@kurta.ru

ООО «Норд клининг»,
Мурманск, 183031, Российская Федерация

Аннотация: В работе решается задача построения аналитической модели интерфейса взаимодействия пользователя с информационной системой, предназначенной для выдачи информации на основании многопараметрических запросов пользователя. Назначением модели является оценка эффективности интерфейса и, соответственно, самого взаимодействия. Для этого, используя пример типовой работы с интерфейсом, вводятся следующие предпосылки (как условия) для моделирования: эффективность интерфейса, его формы и переходы между ними, элементы интерфейса, атомарная эффективность элемента и формы, порядок их расположения и эмпирическая зависимость эффективности, а также постановка задачи проектирования интерфейса. Используя указанные предпосылки, сформирована соответствующая эффективностная модель, позволяющая аналитически, на основании структуры и параметров интерфейса, оценить его результативность, оперативность и ресурсоэкономность. Создан программный прототип, в основе которого лежит модель и с помощью которого проведен ряд экспериментов по трем сценариям, заключающимся в изменении количества форм и элементов интерфейса с целью определения их влияния на общую эффективность; получены соответствующие зависимости в виде гистограмм. Сделаны выводы касательно точности модели, ее теоретической и практической значимости, а также общей сложности проектирования «идеальных» интерфейсов.

Ключевые слова: информационная система, интерфейс, предпосылки, аналитическая модель, эффективность, прототип, эксперимент

Ссылка для цитирования: Курта П.А. Эффективная модель интерфейса взаимодействия пользователя с информационным сервисом запросного типа // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 102–115. DOI:10.31854/1813-324X-2023-9-6-102-115

An Efficient Interface Model of User Interaction with a Query-Type Information Service

Kurta Pavel, expert@kurta.ru

Nord Cleaning Ltd,
Murmansk, 183031, Russian Federation

Abstract: The work solves the problem of an analytical model constructing of the user interaction interface with an information system designed to provide information based on multi-parameter user requests. The model purpose is to evaluate the effectiveness of the interface and, accordingly, the interaction itself. To do this, using an example of typical work with an interface, the following prerequisites (as conditions) for modeling are introduced: the effectiveness of the interface, its forms and transitions between them, interface elements, the atomic efficiency of the element and form, the order of their arrangement and the empirical dependence of efficiency, as well as the formulation of interface design tasks. Using these premises, an appropriate efficiency model has been formed, which allows us to analytically, based on the structure and parameters of the interface, evaluate its effectiveness, operativeness and resource

efficiency. A software prototype was created, which was based on the model and with the help of which a number of experiments were carried out under three scenarios, consisting of changing the number of forms and interface elements in order to determine their impact on overall efficiency; the corresponding dependencies were obtained in the form of histograms. Conclusions are drawn regarding the accuracy of the model, its theoretical and practical significance, as well as the overall complexity of designing "ideal" interfaces.

Keywords: information system, interface, prerequisites, analytical model, efficiency, prototype, experiment

For citation: Kurta P. An Efficient Interface Model of User Interaction with a Query-Type Information Service. *Proceedings of Telecommun. Univ.* 2023;9(6):102–115. DOI:10.31854/1813-324X-2023-9-6-102-115

Введение

Одним из преимуществ повсеместного внедрения информационных технологий стало предоставление человеку возможностей работы с большими объемами гетерогенной информации; например, внедрение концепции Умного дома в быту, обеспечение высокоскоростного доступа к информационным ресурсам практически из любой точки мира, предоставление справочных, обучающих и развлекательных сервисов, а также многое другое. В такой структуре взаимодействия человека с информационной системой (ИС) особое место занимают точки их соединения – интерфейсы (как некоторая совокупность и логика перехода между элементами двухсторонней передачи информации), «удачность» проектирования которых влияет на их эффективность, а также и само взаимодействие [1]. Здесь и далее под эффективностью понимается ее авторская трактовка [2], определяемая тремя классическими показателями: результативностью – отсутствием ошибок в данных, оперативностью – скоростью взаимодействия, и ресурсоэкономностью – сохранностью расходуемых ресурсов. При этом, на характеристики взаимодействия оказывает влияние не только состав интерфейсных элементов, но и состояние человека, особенности обмениваемых данных и ряд других, не всегда очевидных, вещей. Например, выбор человеком страны проживания, как правило, будет осуществлен быстрее и точнее через выпадающий список, чем ручной ввод; и наоборот, указание числа от 1 до 1000 быстрее путем непосредственного ввода с цифрового блока клавиатуры, нежели выбором из списка или «ползунком», хотя точность ввода в обоих случаях будет одинаковой. Также, находящемуся в состоянии усталости человеку комфортнее будет обрабатывать информацию, выводимую порциями – на отдельных экранных формах, а не всю сразу; если же, требуется оперативный анализ больших наборов данных в рамках должностных обязанностей, то целесообразнее предоставлять информацию оператору моментально, а не через последовательно появляющиеся формы. Тем не менее, преобладающее большинство интерфейсов современных программно-аппаратных средств (программ на персональных компьютерах, банкоматах, информационно-электронных

стендах в торговых центрах и т. п.) учитывают лишь общие требования к эргономике интерфейсов, дизайнерские решения отдельных разработчиков или частные методики сравнения и выбора вариантов (как при А/В-тестировании) без рассмотрения остальных участников процесса взаимодействия: заказчика информации – человека, и носителя информации – данных.

Приведенные выше примеры и выводы «очерчивают» предметную область текущего исследования, как – *интерфейсы взаимодействия человека с ИС с позиции их эффективности*. В связи с этим возникает следующее научное противоречие предметной области, сформулированное в виде противопоставления возможностей (т. е. имеющегося состояния дел) и потребностей (т. е. необходимого состояния дел). С одной стороны, используемые и разрабатываемые интерфейсы ИС строятся по достаточно абстрактным схемам (отстраненным от реальности), не учитывающим внешние и внутренние стороны процесса обмена информацией (пользователей, данные, условия применения и т. п.). С другой стороны, одним из основных факторов повышения эффективности интерфейса является учет восприятия информации пользователем [3], на которое влияет, как указывалось, его состояние (психическое, физиологическое и иное), способ ввода и предоставления данных, а также порядок такого предоставления и т. п.

Первым же существенным шагом по разрешению противоречия может стать построение аналитической модели такого взаимодействия (далее – Модель), позволяющей формально оценивать его эффективность – т. е. *эффективностной модели*, что и является задачей текущего исследования [4]. А исходя из того, что при существующем разнообразии целей и задач ИС, создание для них единой Модели является целой проблемой, то ограничимся следующим их подклассом (согласно авторской таксономии). Во-первых, будем рассматривать только ИС, являющиеся информационными сервисами [5] – т. е. ориентированными на решение определенной задачи и обладающими алгоритмами для ее решения. Во-вторых, возьмем лишь информационные сервисы запросного типа – т. е. те, в которых инициатором взаимодействия является человек (например, справочные системы, осу-

ществляющие поиск и предоставление информации по ключевым словам); данный подкласс ИС будем далее сокращенно называть – Сервисом. В-третьих, не будем учитывать функционал интерфейсов, заключающийся в зависимости порядка отображения и состава новых форм на основании ввода значений на предыдущих (исключив тем самым, например, системы с оплатой услуг несколькими способами: банковскими картами с необходимостью ввода всех ее реквизитов, по QR-коду с отображением изображения для сканирования или через систему быстрых платежей с полем ввода номера мобильного телефона) – т. е. логику переходов между ними будем считать линейной, что тем не менее позволит смоделировать огромное количество реально функционирующих ИС. И, в-четвертых, поскольку основным предназначением Сервиса является итоговая выдача запрошенных данных, элементом вывода будем считать только текстовое поле (с его помощью возможно отобразить любую информацию, даже 3D-изображение). Отметим, что помимо данного типа Сервиса, автором выделяется еще два следующих, не рассматриваемых далее: управляющий – который сам является инициатором (например, устройства Умного дома, требующие реакции от человека) и игровой – в котором инициатором взаимодействия являются оба участника (например, любой игровой автомат).

Предпосылки к созданию

Для построения Модели применим следующий *эмпирико-индуктивно-формализационный* методологический подход. Суть его будет заключаться в том, что изначально предложим пример типовой работы пользователя с интерфейсом Сервиса – т. е. будем основываться на данных, полученных исходя из гипотетического эксперимента. Затем по его результатам предложим базовые предпосылки для создания Модели – будет осуществлен переход от частного к общему. И, наконец, систематизируем предпосылки в единую схему работы интерфейса, которая затем подвергнется формализации. А поскольку назначением Модели является оценка эффективности интерфейса, то в ней должны будут отражены механизмы такой оценки.

Предположим, что Сервисом является поиск телефонного номера человека в базе данных по его полному имени, дате рождения и месту проживания. И хотя вариаций интерфейсов даже для такой типовой задачи может быть достаточно много (все элементы расположатся на одной интерфейсной форме или же для каждого элемента используется своя форма и при этом ввод даты рождения можно осуществлять как численно, так и через встроенный календарь), выберем на взгляд автора наиболее гармоничный интерфейс. Для ввода будут использоваться три формы, каждая из которых предназначена для своей группы данных – 1)

имя, фамилия и отчество (ФИО); 2) день, месяц и год рождения; 3) страна и город проживания; соответственно, на каждой форме, кроме последней, должна присутствовать кнопка продолжения работы (для перехода на следующую). В первой группе данных будем использовать текстовый ввод ФИО, во второй – выбор из выпадающего списка соответствующих чисел (дня – от 1 до 31, месяца – от 1 до 12 и года – от 1900 до 2023), а в третьей – выбор из выпадающего списка страны и ввод в текстовое поле города. Для вывода найденной информации будет предназначаться форма с одним текстовым полем в формате «+X (XXX) XXX – XX – XX», где X – цифра телефонного номера.

Процесс единичного решения задачи человека с помощью Сервиса описывается следующей последовательностью действий. Пользователю открывается 1-я форма, элементы которой он должен заполнить. В первый момент пользователем затрачивается определенное количество времени и умственной деятельности на восприятии формы, зависящее, в частности, от общего количества элементов на ней; затем пользователь начинает процесс ввода данных. Из-за необходимости указывать ФИО вручную, с одной стороны, повышается риск ошибки во введенных данных, а с другой – уменьшается время ввода по сравнению с тем, если бы пришлось выбирать каждый компонент (фамилию, имя и отчество) из выпадающего списка (из-за их огромного многообразия).

Оценка психоэмоционального напряжения (ПЭН), вызванного затратами ресурсов человека в случае ручного ввода или выбора из списка, является предметом дискуссии и требует проведения дополнительных исследований; однако очевидно, что оно будет различным. После окончания работы с 1-й формой пользователь должен найти кнопку перехода на 2-ю форму и нажать на нее (некие временные издержки и затрата «нервов»). На этой форме пользователь производит ввод даты рождения с меньшим количеством ошибок, по сравнению с вводом ФИО, хотя расходует больше времени. При этом как по мере перехода к элементам в рамках формы, так и при продвижении к следующим формам все показатели эффективности будут незначительны, но снижаться в связи с нарастанием ПЭН пользователя (поскольку ему необходимо воспринять «картинку», выделить ее компоненты, проанализировать и проинтерпретировать новый вид и суть представляемой информации). Все это будет вести к увеличению количества ошибок при вводе, замедлению работы с элементами, а также ухудшению психоэмоциональное состояние.

Работа с 3-й формой будет сочетать как выбор из выпадающего списка одних значений (страны), так и текстовый ввод других (города), в результате чего предсказание показателей эффективности

становится еще более сложным. На 4-й форме отображается итоговая информация, состоящая из одного значения, вариация чего в общем представляется маловероятной.

Отметим, что конструкция интерфейса, в которой осуществляется переход к новым формам с постепенным заполнением полей, достаточно популярна. Причина этого, вероятно, заключается в субъективном удобстве интерфейса для большинства пользователей. В качестве примера интерфейса можно привести Интернет-сервис по подаче деклараций о налоговом вычете на Web-сайте Федеральной Налоговой Службы (рисунок 1).

Согласно проведенному гипотетическому эксперименту по использованию простого интерфейса поискового Сервиса, эффективность работы пользователя с каждым элементом была различной – для части из них возрастала вероятность ошибок, на другие затрачивалось большее количество времени, а после третьих у пользователя нарастало ПЭН. Сами данные, передаваемые через интерфейс, также оказывали влияние на эффективность – в зависимости от используемых для них элементов. Каждая форма при первом появлении также в той или иной степени затрачивала человеческие ресурсы на осознание представленной на ней информации. При этом разбиение всего интерфейса на множество форм приводило как к появлению дополнительных элементов – кнопок, так и влияло на общее ПЭН от работы (не обязательно в сторону роста напряжения).

Важнейшим выводом из эксперимента является тот факт, что конструирование «идеального» – в смысле максимально эффективного – интерфейса представляется крайне сложно решаемой задачей даже теоретически, поскольку повышение одного показателя эффективности (например, результативности путем замены текстового поля на выпадающий список) в большинстве случаев ведет к снижению других (например, оперативности из-за необходимости выбора из огромного списка нужного элемента и ресурсоэкономности из-за увеличения ПЭН пользователя при таком «скроллинге» списка) [6]. Ситуация усложняется тем, что более дальние формы (с позиции «дальности» на пути продвижения по интерфейсу) и элементы (с позиции «дальности» расположения на форме) начинают снижать показатели эффективности из-за особенностей функционирования человека (в отличие, например, от автомата), что приводит к необходимости учета структуры интерфейса – с позиции его разбиения на формы и размещения на них элементов, и параметров интерфейса – с позиции выбора типов элементов, передаваемых данных и т. п. При этом, большее число форм с одной стороны добавит новые элементы, а с другой – позволит пользователю быстрее подготовиться к работе с ней, сохранив больше ресурсов. Все это говорит об общей сложности проектирования интерфейсов и необходимости в ряде случаев увеличивать не всю эффективность, а лишь отдельные ее показатели – согласно заданным целям и условиям.

Рис. 1. Пример интерфейса Интернет-сайта Федеральной Налоговой Службы

Fig. 1. Example of the Internet Site of the Federal Tax Service Interface

Целесообразным же решением может стать подстройка интерфейса в режиме реального времени – путем указания требований к показателям для конкретных сфер деятельности, задач или групп пользователей [7]. Так, например, в банковской сфере, где в приоритете является точность, интерфейс можно спроектировать при максимизации его результативности; в сфере реагирования на чрезвычайные ситуации – стремясь увеличить оперативность; в сфере социальных услуг (особенно, для пожилых групп пользователей) – повышая ресурсоэкономность; а при размещении Сервисов в общедоступных местах – позволяя настраивать показатели самостоятельно. Касательно последнего можно привести довольно интересную гипотетическую ситуацию, когда пользователь ИС в торговом центре на основании своего текущего состояния (усталости, спешки, знакомства с сервисом и т. п.) сможет точно подстроить интерфейс для работы.

Систематизируем частные выводы из гипотетического эксперимента путем их обобщения в виде предпосылок для построения Модели интерфейсов Сервиса данного – запросного – типа.

Предпосылка 1. Эффективность интерфейса

Эффективность интерфейса представляет собой совокупность трех классических показателей со следующей интерпретацией:

1) ресурсоемкость – мера допущения ошибок при вводе данных (поскольку их вывод при отсутствии дефектов в реализации интерфейса будет всегда корректным);

2) оперативность – быстрота обмена данными между человеком и Сервисом (в основном с точки зрения их ввода);

3) ресурсоэкономность – отсутствие или мера незначительности ПЭН человека при работе с интерфейсом (ресурсы Сервиса считаются несущественными, поскольку обмен и обработка сверхбольших объемов данных через формы или их элементы, требующих запредельных показателей аппаратного обеспечения ИС, выходят за рамки решения типовых задач Сервиса по реагированию на запросы).

Предпосылка 2. Формы интерфейса

Интерфейс состоит из последовательного перехода между формами, содержащими набор элементов; например, таким образом можно группировать ввод ФИО человека, данных о дате и месте рождения, городе проживания, семейном положении, образовании, интересах и т. п.

Предпосылка 3. Переход между формами

Движение по логике интерфейса осуществляется последовательно через все формы с помощью нажатия на соответствующей кнопки; например, в правом нижнем углу с надписью «Далее».

Предпосылка 4. Элементы интерфейса

На каждой форме присутствует последовательность элементов интерфейса для ввода или вывода; например, поля ФИО, как правило, располагаются вертикально друг за другом. Элементы могут типизироваться согласно их предназначению – для ввода или вывода информации, а также принадлежать к некоторому классу, определяющему то, как эта информация передается [8]. Далее будем рассматривать лишь следующие основные классы элементов ввода: текстовое поле, выпадающий список, флаговая кнопка (т. е. CheckBox, *транслит. на русс.* Чекбокс), двунаправленный счетчик и «ползунок»; элемент вывода будет представлен только текстовым полем, естественно, без возможности редактирования.

Предпосылка 5. Атомарная эффективность элемента

Каждый интерфейсный элемент определенного класса обладает собственной атомарной эффективностью взаимодействия с пользователем, зависящей (в грубом, но приемлемом приближении) от типа и размера передаваемых данных [9]. Далее будем рассматривать только наиболее популярные типы, а именно – булевские (состояние из двух значений: «1 и 0», «Да и Нет», «Правда и Ложь», и т. п.), числовые неотрицательные (все значения от 0 до условно большого целого числа) и текстовые (последовательности букв некоторого алфавита).

Предпосылка 6. Атомарная эффективность формы

Каждая интерфейсная форма обладает собственной атомарной эффективностью, определяемой потребностями пользователя в следующих действиях (в аналогичном для Предпосылки 5 приближении): вначале, восприятием отображаемых на форме элементов, на что влияет количество последних; в конце, нажатием на кнопку перехода на следующую форму. Заметим, что влияние кнопки перехода на показатели эффективности можно считать инвариантом для конкретной реализации интерфейса.

Предпосылка 7. Порядок элемента на форме

Показатели атомарной эффективности каждого элемента корректируются в зависимости от дальности расположения на интерфейсной форме – поскольку восприятие более поздних элементов отличается от восприятия более ранних; предположительно, оно ухудшается, хотя формально стоит учитывать любую зависимость.

Предпосылка 8. Порядок формы в интерфейсе

Показатели атомарной эффективности самой формы и всех ее элементов корректируются в зависимости от дальности ее расположения на всем пути логики интерфейса по аналогичным с Предпосылкой 7 причинам.

Предпосылка 9. Эмпирическая зависимость эффективности от порядка

Зависимость эффективности элементов и форм от их порядка представляется сложно определенной теоретическими методами и, поэтому, может быть получена эмпирически (например, с помощью экспериментов по сбору статистических данных на целевых группах пользователей).

Предпосылка 10. Задача проектирования интерфейса

Задача проектирования интерфейса сводится к достижению определенных значений (максимальных или конкретных) для одного или нескольких показателей эффективности путем его структурно-параметрического синтеза (комбинирования разбиения на формы, выбора состава и порядка расположения на них элементов, используемых классов, типов и размеров данных и т. п.); при этом вопросы уязвимости интерфейсов (т. е., например, изначальные ошибки в логике работы Сервиса) [10] в данном контексте не рассматриваются, поскольку выходят за рамки предметной области.

Формализация полученных предпосылок, которые, как хорошо видно, находятся в тесной логической связи, как раз и позволит создать требуемую Модель.

Аналитическая модель

Проведем пошаговую формализацию всех 10 указанных предпосылок к созданию Модели. В качестве обеспечения подобию будем придерживаться следующего стиля записей: нижний индекс – порядковый номер объекта в списке (например, форм); верхний индекс – идентификатор объекта для заранее известного или заданного множества (например, классов); « \sim » – диакритический знак «волна» для выделения функций, вычисляющих результирующие значения по ее параметрам (например, атомарную эффективность по классу элемента, типу и размеру данных); « $\bar{}$ » – диакритический знак «две линии» для выделения процессов, продуцирующих функции (например, атомарную эффективность элемента); «'» и «''» – диакритические знаки «апостроф» и «двойной апостроф» для указания первоначально и вторично уточненного значения (т. е. после применения корректировки к основному и первоначально скорректированному); «:=» – символ определения сложной функциональной сущности (например, функции или задачи).

Предпосылка 1. Эффективность интерфейса.

Аналитическая запись

Эффективность интерфейса (*Efficiency*) имеет следующую запись:

$$\begin{aligned} Efficiency &\equiv \\ &\equiv \langle Potency, Operativeness, ResourceSaving \rangle, \end{aligned}$$

где « $\langle \rangle$ » – кортеж из последовательности следующих показателей эффективности: *Potency* – результативность, *Operativeness* – оперативность, *ResourceSaving* – ресурсоэкономность.

Как будет показано ниже (при расчете комплексной эффективности всего интерфейса), в качестве значений показателей целесообразнее брать некоторые меры в диапазоне $[0,1]$:

$$\begin{cases} Potency \in [0,1] \\ Operativeness \in [0,1] \\ ResourceSaving \in [0,1] \end{cases}.$$

С точки зрения результативности такой диапазон значений является достаточно логичным, поскольку данный показатель определяет меру допущения ошибок. Так, если какой-либо интерфейсный элемент принципиально не позволяет ввести данные (например, для ввода текста используется счетчик чисел), то его результативность всегда будет равна 0; а если для ввода единственного числа выбрана установленная и нередактируемая флаговая кнопка, то результативность будет равна 1 (что, впрочем, не имеет практического смысла). Саму результативность можно оценивать через действия (*Actions*), как отношение количества верных вводов (а формально, и выводов) данных ($Actions^{Valid}$) к количеству всех вводов ($Actions^{All}$):

$$Potency = \frac{Actions^{Valid}}{Actions^{All}}.$$

Подобная ситуация характерна и для ресурсоэкономности – максимально простая работа с элементом или формой (т. е. минимизация ПЭН) делает значение показателя равным 1, а максимальная сложная – 0. Саму ресурсоэкономность можно оценивать по результатам опроса пользователя (т. е. ответа на вопрос «Насколько сложно было работать?») по окончании единичного взаимодействия на основании его собственных ощущений (*Feeling*):

$$\begin{cases} ResourceSaving = \frac{Feeling^{Current}}{Feeling^{Max}}, \\ Feeling^{Current} \in [0, Feeling^{Max}] \end{cases}$$

где $Feeling^{Current}$ – оценка пользователем своего текущего состояния по шкале, где 0 – психоэмоционально максимально затрудненная работа (вариант ответа в опроснике – «Невозможно»); $Feeling^{Max}$ – максимально простая и комфортная (вариант ответа в опроснике – «Элементарно»).

В случае оперативности, определение реальных значений показателя будет несколько иным, поскольку, как правило, быстрота процесса (или скорость) определяется временем работы, на которое делится ее объем (пройденное расстояние, количество движений или просто условная единица). При

таким подходе краевые значения оперативности достижимы лишь в пределе: 0 – в случае бесконечного времени работы с элементом, и бесконечность (формально получаемая при делении на близкое к нулю время) – при моментальном выполнении всех действий.

Привести данный показатель к единичной шкале можно с помощью времени работы ($Time$), которое является минимально возможным ($Time^{Min}$), определяемым эмпирическим путем, и которое будет делиться на текущее затраченное время ($Time^{Current}$):

$$Operativeness = \frac{Time^{Min}}{Time^{Current}}.$$

Предпосылка 2. Формы интерфейса. Аналитическая запись

Полный интерфейс Сервиса ($Interface$) имеет следующую запись:

$$Interface = [Form_1, \dots, Form_i, \dots, Form_N],$$

где «[]» – последовательность форм интерфейса; $Form_i$ – i -я форма; N – количество всех форм.

Поскольку каждая форма состоит из элементов ($Element$), то она имеет следующую запись:

$$Form = [Element_1, \dots, Element_j, \dots, Element_M],$$

где «[]» – последовательность элементов формы; $Element_j$ – j -й элемент; M – количество всех элементов на форме.

Предпосылка 3. Переход между формами. Аналитическая запись

Движения по интерфейсу через формы осуществляются с помощью интерфейсной кнопки перехода ($Button$), имеющей следующую запись:

$$Form_x^{Button} = Button_{x \rightarrow y},$$

где $Form_x^{Button}$ – кнопка на x -й форме; y – индекс формы, на которую по нажатию кнопки происходит переход с x -й; « $x \rightarrow y$ » – обозначение такого перехода между формами.

В условиях ограничений касательно линейности логики переходов между формами, очевидно, что интерфейс состоит из множества следующих кнопок перехода:

$$\begin{aligned} Interface^{Buttons} &= \\ &= \{Button_{1 \rightarrow 2}, \dots, Button_{x \rightarrow x+1}, \dots, Button_{N-1 \rightarrow N}\}, \end{aligned}$$

где «{ }» – множество кнопок.

Данное выражение верно, т. к. с каждой x -й формы возможен переход только на $x + 1$ -ю.

Поскольку последняя форма с выводом результата, очевидно, не содержит кнопки перехода, то количество физических кнопок будет на один меньше количества форм в интерфейсе; для сохранения же подобности и гармоничности формальной записи можно считать, что отсутствие кнопки на последней форме сопоставимо с виртуальной кнопкой $Button_{N \rightarrow N+1}$, которая не влияет на эффективность, т. е. ее эффективность:

$$Efficiency^{Button_{N \rightarrow N+1}} = \{1, 1, 1\}.$$

Отметим, что кнопка завершения работы на последней форме отсутствует, поскольку предполагается, что взаимодействие с Сервисом заканчивается в момент вывода итогового результата.

Предпосылка 4. Элементы интерфейса. Аналитическая запись

Тип элемента интерфейса ($Element^{Type}$) имеет следующую запись, соответствующую способности осуществлять двухсторонний обмен между пользователем и Сервисом:

$$Element^{Type} \in \{Element^{\rightarrow}, Element^{\leftarrow}\},$$

где «{ }» – множество типов элементов; $Element^{\rightarrow}$ и $Element^{\leftarrow}$ – тип элемента, предназначенный для ввода и вывода данных, соответственно (символ верхнего индекса-стрелки выбран, исходя из направления передачи данных – от пользователя к Сервису).

Также элементы ввода могут принадлежать различным классам (являющимся подмножеством типа), где $Element^{Class \rightarrow}$ – класс элемента ввода данных; а другие верхние индексы для $Element$ соответствуют классам следующим образом: $TextBox$ – текстовое поле; $DropDownList$ – выпадающий список; $CheckBox$ – флаговая кнопка; $Spinner$ – двунаправленный счетчик; $Slider$ – «ползунок» (1).

Согласно указанным ограничениям, элементы вывода могут относиться только к одному классу:

$$Element^{Class \leftarrow} \in \{Element^{TextBoxRO}\},$$

где верхний индекс $TextBoxRO$ соответствует текстовому полю без возможности редактирования (постфикс RO – аббр. от англ. *Read Only*; пер. на русс. Только чтение).

Исходя из того, что множества классов элементов для различных направлений передачи данных не пересекаются, их можно объединить в один класс, опуская тем самым (здесь и далее) указание типа элемента:

$$Element^{Class} = Element^{Class \rightarrow} \cup Element^{Class \leftarrow},$$

где \cup – операция объединения множеств классов элементов.

$$Element^{Class \rightarrow} \in \{Element^{TextBox}, Element^{DropDownList}, Element^{CheckBox}, Element^{Spinner}, Element^{Slider}\}. \quad (1)$$

Предпосылка 5. Атомарная эффективность элемента. Аналитическая запись

Атомарная эффективность интерфейсного элемента ($Efficiency_{Element_j}$) имеет вид (2), где $Efficiency^{Element}(\dots)$ – функция вычисления атомарной эффективности элемента согласно его классу ($Element^{Class}$) и свойствам данных ($Data$), состоящим из типа ($Data^{Type}$) и размера ($Data^{Size}$), для передачи которых он используется.

Тип данных записывается следующим образом:

$$Data^{Type} \in \{Data^{Boolean}, Data^{Unsigned}, Data^{Text}\},$$

где « $\{ \}$ » – множество типов данных; а индексы для $Data$ соответствуют типам следующим образом: *Boolean* – булевский; *Unsigned* – целое беззнаковое число; *Text* – текст.

Предпосылка 6. Атомарная эффективность формы. Аналитическая запись

Атомарная эффективность интерфейсной формы ($Efficiency_{Form}$) имеет следующую запись:

$$Efficiency_{Form} = Efficiency^{Form}(M_{Form}, Button_{x \rightarrow x+1}),$$

где $Efficiency^{Form}(\dots)$ – функция вычисления атомарной эффективности формы согласно количеству элементов на ней (M_{Form}) и кнопке перехода на следующую форму (для x -й формы – $Button_{x \rightarrow x+1}$).

Исходя из того, что влияние кнопки на показатели эффективности считается инвариантом в рамках одного интерфейса, будет верно следующее уравнение (для любой x -ой формы с одинаковым количеством элементов):

$$Efficiency^{Form}(M, Button_{x-1 \rightarrow x}) = Efficiency^{Form}(M, Button_{x \rightarrow x+1}),$$

т. е. атомарные эффективности форм будут отличаться только при разном количестве элементов на них; в ином случае, следуя уравнению, они равны.

Предпосылка 7. Порядок элемента на форме. Аналитическая запись

Эффективность интерфейсного элемента после коррекции ($Efficiency'_{Element}$), которая учитывает порядок его расположения на форме, имеет вид (3), где $Efficiency^{Correction}_{Element}(\dots)$ – функция вычисления скорректированной эффективности элемента по ее атомарному значению и порядковому индексу (*Index*).

Предпосылка 8. Порядок формы в интерфейсе. Аналитическая запись

Эффективность элемента ($Efficiency''_{Element}$) и формы ($Efficiency'_{Form}$) после коррекции, которая учитывает порядок формы в логике интерфейса, имеет вид (4), с использованием единой функции.

В выражении (4) $Efficiency^{Correction}_{Form}(\dots)$ – функция вычисления скорректированных эффективностей элемента и формы по первоначально скорректированной (согласно Предпосылке 7) и атомарной, соответственно; *Index* – порядковый индекс формы в логике интерфейса, согласно которому осуществляется коррекция. Таким образом, эффективность формы подвергается однократной коррекции по порядковому индексу, а эффективность элемента – двухкратной коррекции по паре порядковых индексов.

Предпосылка 9. Эмпирическая зависимость эффективности от порядка. Аналитическая запись

Поскольку каждая из введенных функций определяет достаточно сложную зависимость между своими параметрами и выходным значением (например, оперативность работы пользователя с двунаправленным списком вряд ли будет линейно зависеть от количества отображаемых им значений), то способ определения такой взаимосвязи может быть определен статистически, путем проведения соответствующих экспериментов. Всего насчитываются 4 их группы, в рамках каждой из которых определяются следующие зависимости: 1) атомарная эффективность элемента от комбинации его класса, а также типов и размеров данных (см. Предпосылку 5); 2) атомарная эффективность формы от количества элементов на ней и различных кнопок перехода (см. Предпосылку 6); 3) скорректированная эффективность элемента от его порядка на форме (см. Предпосылку 7); 3) скорректированная эффективность формы от ее порядка в логике интерфейса (см. Предпосылку 8). Данные группы экспериментов (*Experiment*) могут быть записаны в формальном виде (5), где $Experiment$ – процесс получения функции (т. е. алгоритма вычисления значений по параметрам согласно некоторым их соотношениям) для вычисления или корректировки эффективностей, верхний индекс которого определяет назначение функции: *Element* – атомарная эффективность элемента, *Form* – атомарная эффективность формы, *ElementCorrection* – корректировка атомарной эффективности элемента, *FormCorrection* – корректировка атомарной эффективности формы; $\{Subject_i\}$ – множество i -ых субъектов (участников эксперимента), согласно процессу взаимодействия с интерфейсом которых вычисляются показатели эффективности; $\{Element^{Class}_i\}$ – множество i -ых классов элементов (в количестве шести); $\{Date^{Type}_i\}$ – множество i -ых типов данных (в количестве трех); $\{Date^{Size}_i\}$ – множество i -ых размеров данных (например, от 1 до выбранного разумного предела с увеличением значения в 10 раз); $\{M_i\}$ – множество i -ых значений количества элементов на форме (также, от 1 до выбранного разумного предела с шагом 1); $\{Button_i\}$ –

множество i -ых вариаций кнопки перехода (без потери точности достаточно взять один вариант, стандартный для операционной системы интерфейса); $\{Index_j\}$ и $\{Index_i\}$ – множество j -ых индексов положения элемента на типовой форме и i -ых положения формы в логике интерфейса (например, от 1 до некоторого разумного предела с шагом 1).

Результаты экспериментов могут иметь вид зависимостей (аналитической или табличной), которые будут служить основой для вычисления соответствующих функций; например, если зависимость атомарной эффективности формы от количества элементов на ней имеет логистический вид, то в результате проведения эксперимента будут получен соответствующий набор точек на плоскости, сглаженная линия по которым и ляжет в основу вычислений соответствующей функции.

Каждый же эксперимент (*Experiment*, т. е. без указания верхнего индекса для назначения итоговой функции), очевидно, для одного набора параметров позволит получить сразу 3 зависимости (*Dependence*) – для всех показателей эффективности, где $Dependence^{Potency}$, $Dependence^{Operativeness}$ и $Dependence^{ResourceSaving}$ – зависимость от параметров эксперимента (и, следовательно, функции) результативности, оперативности и ресурсоэкономности соответственно (6).

Предпосылка 10. Задача проектирования интерфейса. Аналитическая запись

Общая эффективность для интерфейса, содержащего определенную структуру и параметры форм, кнопок перехода, элементов и их данных, имеет вид (7), где индекс оператора умножения i от 1 до N соответствует перебору всех форм интерфейса;

индекс j от 1 до M_i – перебору всех элементов на i -й форме с их количество M_i ; пара « i, j » – комплексному индексу, обозначающему j -й элемент (и его данные) на i -й форме.

Суть определения общей эффективности состоит в перемножении двух следующих выражений. В первом множителе производится вычисление атомарных эффективностей всех форм интерфейса с применением функции их корректировки согласно порядку формы. Во втором множителе производится вычисление атомарной эффективности каждого элемента, расположенного на каждой форме, согласно его классу, типу данных и их размеру. Затем, эффективность элемента корректируется с учетом его порядка на каждой форме, а также порядка самой формы в логике интерфейса. Исходя из этих выражений, выбирая разбиение элементов по формам, классы элементов, учитывая передаваемые данные, изменяя порядок обоих сущностей, а также упорядочивая формы и элементы на них, можно «играться» итоговыми показателями эффективности – т. е. производить ее многокритериальную оптимизацию.

Необходимо указать, что умножение атомарных, скорректированных, общих и иных эффективностей заключается в поэлементном умножении ее показателей, т. е.:

$$\begin{aligned} Efficiency^X \times Efficiency^Y &\equiv \\ &Potency^X \times Potency^Y, \\ &\equiv \langle Operativeness^X \times Operativeness^Y, \rangle, \\ &ResourceSaving^X \times ResourceSaving^Y \end{aligned}$$

где x и y – верхние индексы, относящиеся к перемножаемым эффективностям.

$$\left\{ \begin{aligned} Efficiency_{Element} &= Efficiency_{Element}^{Element} (Element^{Class}, Data), \\ Data &\equiv \langle Data^{Type}, Data^{Size} \rangle \end{aligned} \right. \quad (2)$$

$$Efficiency'_{Element} = Efficiency_{Correction}^{Element} (Efficiency_{Element}, Index), \quad (3)$$

$$\left\{ \begin{aligned} Efficiency''_{Element} &= Efficiency_{Correction}^{Form} (Efficiency'_{Element}, Index) \\ Efficiency'_{Form} &= Efficiency_{Correction}^{Form} (Efficiency_{Form}, Index) \end{aligned} \right. \quad (4)$$

$$\left\{ \begin{aligned} Efficiency^{Element} &:= \overline{Experiment}^{Element} (\{Subject_i\}, \{Element_i^{Class}\}, \{Date_i^{Type}\}, \{Date_i^{Size}\}) \\ Efficiency^{Form} &:= \overline{Experiment}^{Form} (\{Subject_i\}, \{M_i\}, \{Button_i\}) \\ Efficiency_{Correction}^{Element} &:= \overline{Experiment}^{ElementCorrection} (\{Subject_i\}, \{Index_j\}) \\ Efficiency_{Correction}^{Form} &:= \overline{Experiment}^{FormCorrection} (\{Subject_i\}, \{Index_i\}) \end{aligned} \right. \quad (5)$$

$$\langle Dependence^{Potency}, Dependence^{Operativeness}, Dependence^{ResourceSaving} \rangle = \overline{Experiment} (...), \quad (6)$$

$$\begin{aligned} Efficiency^{Interface} &= \prod_{i=1}^N Efficiency_{Correction}^{Form} (Efficiency_{Form}^{Form} (M_i, Button_{i \rightarrow i+1}), i) \times \\ &\times \prod_{i=1}^N \prod_{j=1}^{M_i} Efficiency_{Correction}^{Form} (Efficiency_{Correction}^{Element} (\\ &Efficiency_{Element}^{Element} (Element_{i,j}^{Class}, \langle Data^{Type}, Data^{Size} \rangle_{i,j}), j), i) \end{aligned} \quad (7)$$

Обоснованность применения умножений (вместо типичного суммирования) при расчете общей эффективности заключается в том, что, как указывалось, показатели отражают степень достижения максимальных нормированных значений (т. е. единицы), а каждый элемент, форма или кнопка эти значения могут уменьшать. Например, если на форме присутствует 3 элемента, ухудшающих безошибочность ввода данных на 10 %, то результативность каждого равна 0.9, а результативность всей группы элементов в случае суммирования будет $0.9 + 0.9 + 0.9 = 2.7$, что, очевидно, неверно; правильным будет перемножение результативностей, т. е. что даст значение показателя – $0.9 \times 0.9 \times 0.9 = 0.729$.

Таким образом, общую задачу проектирования интерфейсов (*Task*), исходя из их целевой (*Target*) эффективности, можно записать в виде (8), где *Interface*() – интерфейс, спроектированный на

основании множества *i*-ых форм $\{Form_i\}$ (естественно, с нужным количеством кнопок перехода) и множества *j*-ых элементов на каждой *i*-ой форме $\{Element_{i,j}\}$, каждый из которых имеет соответствующий класс $Element_{i,j}^{class}$; « \Rightarrow » – суть задачи в виде указания искомым параметрам (слева), в результате которых должны выполняться условия (справа); « \rightarrow » – оператор стремления (в пределе, равенства) эффективности к целевому значению (*Target*), состоящему из совокупности 3-х показателей – $Target^{Operativeness}$, $Target^{ResourceSaving}$ и $Target^{Potency}$, соответствующих показателям результативности, оперативности и ресурсоэкономности; $Target^{Value}$ – значение, которое может принимать показатель целевого значения эффективности и которое может быть следующим: *Certain* – точное число, *Max* – максимально допустимое число, *Any* – произвольное число (означающее, что при решении задачи данный показатель не должен учитываться).

$$\left\{ \begin{array}{l} Task := \\ Interface(\{Form_i\}, \{Element_{i,j}\}, \{Element_{i,j}^{class}\}) \Rightarrow \\ Efficiency^{Interface} \rightarrow Target \\ Target \equiv \langle Target^{Potency}, Target^{Operativeness}, Target^{ResourceSaving} \rangle \\ Target^{Value} \in \{Certain, Max, Any\} \end{array} \right. \quad (8)$$

Так, решение задачи проектирования интерфейса для банковской сферы, когда важна лишь точность, может осуществляться при следующих целевых значениях показателей:

$$Target \rightarrow \langle Max, Any, Any \rangle.$$

При реагировании на чрезвычайные ситуации, требующем высокой оперативности [11], целевые показатели будут следующими:

$$Target \rightarrow \langle Any, Max, Any \rangle.$$

А в сфере социальных услуг для пожилых групп [12] целевыми показателями могут быть заданы, как:

$$Target \rightarrow \langle Any, Any, Max \rangle,$$

поскольку тут уже важна ресурсоэкономность. Если же при этом необходима и точность передачи данных от посетителей, то:

$$Target \rightarrow \langle Max, Any, Max \rangle,$$

что, впрочем, может быть недостижимо из-за наличия многоцелевой оптимизации (если только результативность и ресурсоэкономность не имеют независимую максимизацию за счет снижения скорости обслуживания очереди); для случая же, когда необходимо пользователю предоставить возможность самостоятельно выбирать эффективность интерфейса с Сервисом, будет:

$$Target \rightarrow \langle Certain, Certain, Certain \rangle,$$

т. е. значения показателей задаются явно (как вводом конкретных чисел, так и «ползунками» для указания их отношения друг между другом).

Эксперимент

Для проверки общей работоспособности Модели, обоснования ее необходимости (из-за нетривиальности зависимостей между структурой/параметрами интерфейса и его эффективностью), а также получения предварительных оценок, был создан программный прототип для проведения соответствующих экспериментов (далее – Прототип); его разработка велась в среде Microsoft Visual Studio Community 2019 на языке программирования C#.

Работа прототипа состоит из следующей последовательности шагов.

Шаг 1. Строится интерфейс путем указания его форм, элементов, их взаимосвязей и параметров. В качестве заготовок интерфейса может быть взята стандартная конфигурация или сгенерирована группа новых согласно заданным условиям (например, для определенного числа форм и элементов на них).

Шаг 2. Задаются инвариантные параметры интерфейса, определяющие функции корректировки и атомарной эффективности (например, для кнопок перехода).

Шаг 3. Создается Модель согласно построенному интерфейсу и его инвариантным параметрам.

Шаг 4. Производится вычисление общей эффективности интерфейса с выводом итогового и промежуточных результатов на консоль.

Эксперименты были упрощены путем указания всех инвариантных эффективностей, равных значению некоторой эталонной – $\{0.9999, 0.9999, 0.9999\}$ (т. е. близкой, но не равной, к идеальной). Сами же инварианты состояли из следующих эффективностей – для элементов всех классов, данных всех типов, формы и кнопки перехода. Учет количества элементов на форме производился путем умножения эффективности формы на эталонную столько раз, сколько было на ней элементов; учет размера данных элемента осуществлялся аналогичным образом, но количество умножений бралось равным разрядности размера (например, для разрядности 1000 равно 3); а для корректировки эффективностей количество умножений равнялось соответствующему индексу. Также считалось, что все элементы относятся к классу текстовых полей и предназначены для отображения 1 символа.

Во всех экспериментах элементы равномерно размещались на формах, в случае некратности количества первых (N) ко вторым (M) «лишние» элементы размещались по формам случайным образом. Например, 10 элементов по 3 формам могли разместиться следующим образом: 3, 4 и 3. Такое распределение можно увидеть в следующем отладочном выводе Прототипа (в формате JSON).

```
{
  'Name': 'Simple interface by N = 3 and M = 10',
  'Forms': [ {
    'Name': 'Form 1',
    'Button': 'Next 1->2',
    'Elements': [{
      'Name': 'Element 1.1',
      'Class': 'TextBox',
      'Data': '<Text, 1>',
    }, {
      'Name': 'Element 1.2',
      'Class': 'TextBox',
      'Data': '<Text, 1>',
    }, {
      'Name': 'Element 1.3',
      'Class': 'TextBox',
      'Data': '<Text, 1>',
    }
  ]
}, {
  'Name': 'Form 2',
  'Button': 'Next 2->3',
  'Elements': [{
    'Name': 'Element 2.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 2.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 2.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
}, {
  'Name': 'Form 3',
  'Button': 'Next 3->4',
  'Elements': [{
    'Name': 'Element 3.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
}
]
```

```
{
  'Name': 'Element 2.4',
  'Class': 'TextBox',
  'Data': '<Text, 1>',
},
], {
  'Name': 'Form 3',
  'Button': 'Next 3->4',
  'Elements': [{
    'Name': 'Element 3.1',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.2',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }, {
    'Name': 'Element 3.3',
    'Class': 'TextBox',
    'Data': '<Text, 1>',
  }
},
],
}
```

Были проведены эксперименты по следующим сценариям построения интерфейсов.

Сценарий 1. Перебор количества форм интерфейса ($N = 1..100$) при неизменном количестве его элементов ($M = 100$). В рамках эксперимента было взято $M = 100$ элементов, которые равномерно размещались на формах, количество N которых менялось от 1 до 100.

Вычисление эффективности интерфейса такой конфигурации для различного количества форм N представлено в виде гистограммы ее показателя на рисунке 2а; показатель является любым из трех, поскольку, согласно условиям эксперимента, они равнозначны.

Гистограмма имеет возрастающее значение показателя эффективности с достижением пика 0.854697349 при количестве форм, равных 9 (отмечено красной линией (см. рисунок 2а) и соответствует наивысшей высоте столбца), с последующим его уменьшением. Таким образом, в сценарии существует некоторое оптимальное соотношение количества форм и элементов на них, при котором показатель эффективности будет максимальным.

Сценарий 2. Перебор количества элементов интерфейса ($M = 1..100$) при неизменном количестве его форм ($N = 100$). В рамках эксперимента было взято $N = 100$ форм, на которых равномерно размещались элементы, количество M которых менялось от 1 до 100.

Вычисление эффективностей интерфейса такой конфигурации для различного количества элементов M представлено в виде гистограммы ее показателя на рисунке 2б. В отличие от гистограммы на рисунке 2а, данная характеризуется уменьшением показателя эффективности при увеличении элементов. Максимальное значение показателя (отмечено красной линией) в эксперименте было достигнуто для 1 элемента и равнялось 0.5911850000251.

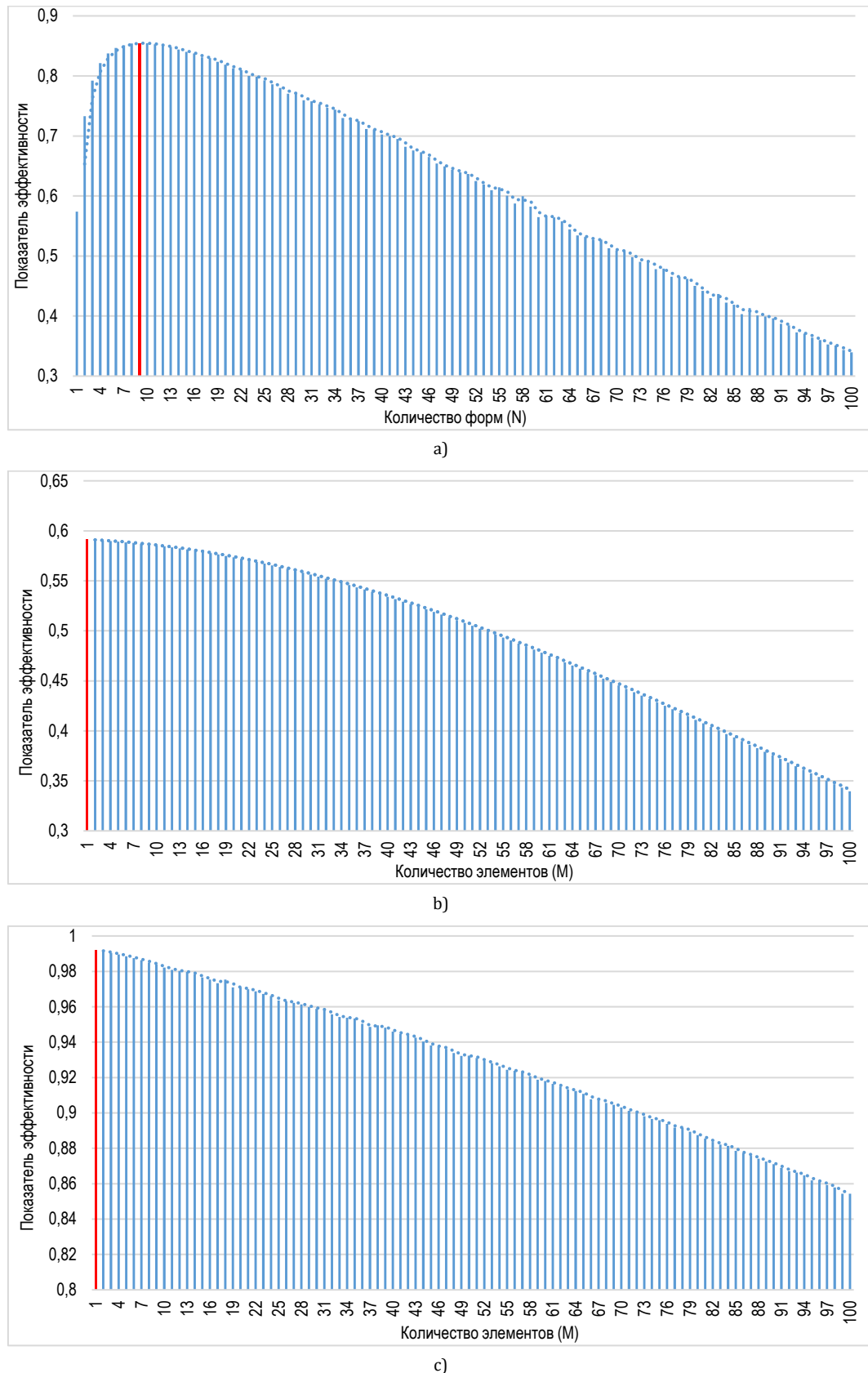


Рис. 2. Зависимость показателя эффективности интерфейса для равномерного распределения 100 элементов по переменному количеству форм (а), переменного количества элементов по 100 формам (б) и по 10 формам (с)

Fig. 2. Dependence of the Interface Efficiency Indicator for Uniform Distribution of 100 Elements across a Variable Forms Number (a), a Variable Number Elements across 100 Forms (b) and 10 Forms (c)

Результат эксперимента по Сценарию 2 является закономерным, поскольку любое увеличение количества элементов (гипотетически, не зависимо от количества форм) ведет к уменьшению эффективности интерфейса; естественно, при решении задачи проектирования интерфейса данным параметром «играться» нельзя, поскольку он определяется неизменной логикой работы Сервиса.

Сценарий 3. Перебор количества элементов интерфейса ($M = 1..100$) при неизменном количестве его форм ($N = 10$). В рамках эксперимента было взято $N = 10$ форм, на которых равномерно размещались элементы, количество M которых менялось от 1 до 100.

Вычисление эффективностей интерфейса такой конфигурации для различного количества элементов M представлено в виде гистограммы ее показателя на рисунке 2с. Аналогично гистограмме на рисунке 2б, максимальное значение показателя эффективности (отмечено красной линией) в эксперименте было достигнуто для 1 элемента и равнялось 0.991932315. Результат аналогичен Сценарию 2 и доказывает, что уменьшение количества элементов более существенно, чем изменение количества форм.

Результаты экспериментов позволяют сделать следующие выводы. Во-первых, Модель достаточно точно отражает особенности интерфейса, поскольку получаемая эффективность существенно зависит от задаваемых параметров. Во-вторых, Модель не носит исключительно теоретический характер, поскольку может быть реализована с помощью программных средств. И, в-третьих, расчет эффективности интерфейса даже при условии сильных допущений является не тривиальной задачей, а ее максимизация требует проведения определенных вычислений или экспериментов.

Заключение

Работа посвящена задаче построения аналитической модели, позволяющей оценивать эффективность интерфейсов взаимодействия пользователя с информационным сервисом запросного типа –

т. е. иницируемым человеком и представляющим собой последовательность форм с вводимыми данными, на основании которых система выдает результат. В интересах этого, на основании типового примера такого взаимодействия выдвинуты 10 предпосылок к построению модели. Каждая из предпосылок переведена в формализованный вид, что позволило создать единую систему, отражающую взаимосвязь форм и элементов интерфейса, их структуры и параметров, а также являющуюся базисом для вычисления общей эффективности интерфейса.

Основным результатом исследования является эффективностная модель интерфейса информационного сервиса запросного типа.

Новизна результатов заключается в авторской модели оценки эффективности интерфейса информационной системы определенного типа, имеющей полностью формальный вид. Кроме того, впервые предложены 10 предпосылок к созданию моделей, которые могут быть адаптированы для достаточно большого класса интерфейсов.

Теоретическая значимость исследования заключается в расширении аппарата формализации и оценивания интерфейсной части информационного взаимодействия, изначально характеризующейся сравнительно высокой степенью субъективности.

Практическая значимость состоит в готовности эффективностной модели к программной реализации, что было продемонстрировано с помощью эксперимента по трем сценариям; полученные в результате зависимости показателей эффективности интерфейса от его структуры и параметров также несут определенную практическую ценность для архитекторов интерфейсов.

Продолжением исследования должна стать разработка полноценного средства моделирования интерфейса информационного сервиса запросного типа, а также проведение большего числа экспериментов.

Список источников

1. Израйлов К.Е., Левшун Д.С., Чечулин А.А. Модель классификации уязвимостей интерфейсов транспортной инфраструктуры «умного города» // Системы управления, связи и безопасности. 2021. № 5. С. 199–223. DOI:10.24412/2410-9916-2021-5-199-223
2. Курта П.А. Взаимодействие пользователя с информационной системой. Часть 3. Оценка эффективности // Известия СПбГЭТУ ЛЭТИ. 2021. № 4. С. 58–72.
3. Самарин М.А., Максимов А.В., Вострых А.В. Повышение эффективности информационных систем типа электронные библиотеки // Природные и техногенные риски (физико-математические и прикладные аспекты). 2023. № 1 (45). С. 84–93.
4. Буйневич М.В., Курта П.А. Методология исследования метода оптимизации информационного взаимодействия в аспекте решения задачи пользователя // Информационные технологии и телекоммуникации. 2019. Т. 7. № 4. С. 50–58. DOI:10.31854/2307-1303-2019-7-4-50-58
5. Ломоносова Т.С. Информационный сервис важность информационных услуг в жизни человека // Теория и практика современной науки. 2015. № 2(2). С. 87–88.

6. Вострых А.В. Метод оценки эффективности графических пользовательских интерфейсов программных продуктов // Приборы и системы. Управление, контроль, диагностика. 2022. № 10. С. 19-28. DOI:10.25791/pribor.10.2022.1365
7. Ярошенко А.Ю. Предпосылки к необходимости непрерывного ранжирования требований пожарной безопасности // Национальная безопасность и стратегическое планирование. 2021. № 3(35). С. 100–105. DOI:10.37468/2307-1400-2021-3-100-105
8. Любчикова О.В. Типы интерфейсов и виды элементов // Новая наука: От идеи к результату. 2016. № 12-3. С. 121–124.
9. Трегубов А.С. Разработка адаптивных контекстозависимых интерфейсов с использованием онтологических моделей // Кибернетика и программирование. 2017. № 6. С. 50–56. DOI:10.25136/2306-4196.2017.6.24747
10. Izrailov K., Levshun D., Kotenko I., Chechulin A. Classification and analysis of vulnerabilities in mobile device infrastructure interfaces // Communications in Computer and Information Science. 2022. Vol. 1544. PP. 301–319. DOI:10.1007/978-981-16-9576-6_21
11. Антюхов В.И., Остудин Н.В., Ярошенко А.Ю., Черных А.К. Информационная потребность должностных лиц Центров управления в кризисных ситуациях (ЦУКС) МЧС России // VII Международная научно-практическая конференция «Сервис безопасности в России: опыт, проблемы, перспективы. Обеспечение безопасности при чрезвычайных ситуациях» (Санкт-Петербург, Российская Федерация, 24 сентября 2015). Санкт-Петербург: Санкт-Петербургский университет ГПС МЧС России, 2015. С. 70–71.
12. Грушко Ю.В. Аппаратно-программный комплекс аугментативной системы коммуникации на основе технологии Eyetracking // Вестник КРАУНЦ. Физико-математические науки. 2019. Т. 27. № 2. С. 55–73. DOI:10.26117/2079-6641-2019-27-2-55-73


References

1. Izrailov K.E., Levshun D.S., Chechulin A.A. Vulnerability Classification Model for Smart City Transport Infrastructure Interfaces. *Systems of Control, Communication and Security*. 2021;5:199–223. DOI:10.24412/2410-9916-2021-5-199-223
2. Kurta P.A. Interaction of the User with the Information System. Part 3. Efficiency Evaluation. *Proceedings of Saint Petersburg Electrotechnical University Journal*. 2021;4:58–72.
3. Samarin M.A., Maximov A.V., Vostrikh A.V. Improving the Efficiency of Information Systems Such as Electronic Libraries. *Natural and Man-Made Risks (Physico-Mathematical and Applied Aspects)*. 2023;1(45):84–93.
4. Buinevich M., Kurta P. Research Methodology Of The Method Of Optimization Of Information Interaction In The Aspect Of Solving The User's Problem. *Telecom IT*. 2019;7(4):50–58. DOI:10.31854/2307-1303-2019-7-4-50-58.
5. Lomonosova T.S. Information Service the Importance of Information Services in Human Life. *Teoriya i praktika sovremennoy nauki*. 2015;2(2):87–88.
6. Vostrykh A.V. A Method for Evaluating the Effectiveness of Graphical User Interfaces of Software Products. *Instruments and Systems: Monitoring, Control, and Diagnostics*. 2022;10:19–28. DOI:10.25791/pribor.10.2022.1365
7. Yaroshenko A. Background to the Necessity of Continuous Ranking of Fire Safety Requirements. *National Security and Strategic Planning*. 2021;3(35):100–105. DOI:10.37468/2307-1400-2021-3-100-105
8. Lyubchikova O.V. Development of adaptive context-sensitive interfaces using ontological models. *Novaya nauka: Ot idei k rezul'tatu*. 2016;12-3:121–124.
9. Tregubov A.S. Development of adaptive context-sensitive interfaces using ontological models. *Cybernetics and programming*. 2017;6:50–56. DOI:10.25136/2306-4196.2017.6.24747
10. Izrailov K., Levshun D., Kotenko I., Chechulin A. Classification and analysis of vulnerabilities in mobile device infrastructure interfaces. *Communications in Computer and Information Science*. 2022;1544:301–319. DOI:10.1007/978-981-16-9576-6_21
11. Antyukhov V.I., Ostudin N.V., Yaroshenko A.YU., Chernykh A.K. Information needs of officials of the Crisis Management Centers of the Ministry of Emergency Situations of Russia. *Proceedings of the VII International Scientific and Practical Conference "Security Service in Russia: Experience, Problems, Prospects. Ensuring Safety in Emergency Situations"*, 24 September 2015, St. Petersburg, Russian Federation. St. Petersburg: Saint-Petersburg State Fire Service University of the Ministry of RF for Civil Defense, Emergencies and Elimination of Consequences of Natural Disasters Publ.; 2015. p.70–71.
12. Grushko Y.V. Hardware and Software Complex of Augmentation Communication System on the Basis of Eyetracking Technologys. *Bulletin KRASEC. Physical and Mathematical Sciences*. 2019;27(2):55–73. DOI:10.26117/2079-6641-2019-27-2-55-73

Статья поступила в редакцию 01.12.2023; одобрена после рецензирования 08.12.2023; принята к публикации 11.12.2023.

The article was submitted 01.12.2023; approved after reviewing 08.12.2023; accepted for publication 11.12.2023.

Информация об авторе:

КУРТА | технический директор ООО «Норд клининг»
Павел Андреевич |  <https://orcid.org/0009-0005-6073-8626>