

УДК 004.5

П. А. Курта

*Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М. А. Бонч-Бруевича*

Взаимодействие пользователя с информационной системой. Часть 3. Оценка эффективности*

Эта статья – третья из цикла, посвященного вопросам взаимодействия пользователя с информационной системой. Рассматривается задача расчета и оценки эффективности взаимодействий. Для этого предлагается отнести группу сравниваемых информационных сервисов (реализации интерфейса взаимодействия и информационной системы) к некоторому классу, для которого построена модель взаимодействий. Затем в терминах модели можно вычислить показатели эффективности – результативность как гарантированность получения результата пользователем; оперативность как быстроту ввода и получения данных; ресурсоэкономность как умеренность когнитивной нагрузки на человека при работе. Полученные значения показателей могут непосредственно использоваться для сравнения взаимодействий. Выделяются 3 класса распространенных информационных сервисов: запросный, управляющий и игровой. Для первого из них, как достаточно простого для исследования, строится аналитическая модель взаимодействия, а также предлагаются 3 схемы реализации, отличающиеся плотностью заполнения интерфейсных форм полями ввода/вывода; выводятся формулы для вычисления показателей. Проводится эксперимент по расчету показателей эффективности взаимодействий в каждой из схем для различного числа интерфейсных элементов. Приводятся значения показателей, а также среднеквадратичной эффективности по ним и их парам в табличном и графическом виде; рассматривается ранжирование эффективности схем по Парето. Анализируются полученные результаты и делаются выводы, которые подтверждают обоснованность полученных формул. Указываются направления дальнейшего исследования.

Информационная система, интерфейс, эффективность, показатели, модель

Глобальная информатизация всех сфер жизнедеятельности привела к появлению множества человеко-машинных взаимодействий (далее – Взаимодействие) с информационными системами (далее – Система) [1], приводя к появлению целого направления человеко-ориентированных информационных сервисов (далее – Сервис), например для поиска в Интернете, управления процессами, исследования и оптимизации стратегий поведения. Вследствие этого возникла необходимость оценки эффективности Взаимодействий как для совершенствования новых Сервисов [2], [3], так и для проектирования новых [4], [5]. Большая вариативность реализаций интерфейсов Взаимодействий, разнообразие решаемых задач, ориентированность на различные группы пользователей сделали такую оценку крайне сложной. Одна же из основных причин снижения эффективности (что было показано в предыдущих двух частях цикла статей [6], [7]) заключает-

ся в наличии ряда дефектов, негативно сказывающихся на ее показателях. В продолжение начатой логики исследования далее будет частично решена задача расчета и сравнения эффективности Взаимодействий для Сервисов некоторого класса, имеющих ряд упомянутых дефектов. Обобщение способа оценки на все множество Взаимодействий – проблема сверхсложная, требует отдельного научного исследования и в данной статье рассматриваться не будет.

Понятие эффективности. В научных работах описываются различные способы оценки эффективности Взаимодействий и используемых в них интерфейсов. Часть из них выделяет составляющие интерфейса, имеющие собственные критерии оценки – аппаратную, программную и визуальную [8]. Другие определяют эффективность по количеству пользователей и качеству их обучения при дистанционном образовании [9]. Некоторые же полностью полагаются на соответствие принятым стандартам [10]. В ряде случаев предлагается проводить имитационные и натурные

* Окончание. Начало № 8–9, 10/2020.

эксперименты [11], в других для оценки могут применяться модели качества программного обеспечения [12]. Однако можно с уверенностью утверждать, что к основным требованиям при работе в Сервисах с позиции пользователей относятся следующие:

1. Гарантированная возможность решения требуемой задачи – показатель того, что задача в принципе решается в рамках Взаимодействия с Системой; так, например, большое количество разветвлений в переходах между точками сценария Взаимодействия (в особенности бесполезных) будет затруднять для пользователя получение требуемого результата. В данном случае под сценарием (далее – Сценарий) понимается логика работы пользователя в процессе Взаимодействия с Системой, заложенная в интерфейс (например, порядок отображения экранных форм с графическими элементами).

2. Быстрота ввода и получения данных – по сути, скорость взаимодействия пользователя с интерфейсом, включая как работу с информацией на формах, так и переключение между ними; например, неудачный выбор интерфейсных элементов (текстовое поле для ввода бинарных значений, выпадающий список для выбора слов из словаря, звуковое уведомление для вывода идентификационного номера и пр.) будет замедлять работу с системой, не давая практически никаких преимуществ.

3. Низкая когнитивная нагрузка на человека – показатель простоты восприятия информации (получаемый, например, из метрики понятности [13]), как запрашиваемой у пользователя, так и предоставляемой ему; например, отображение огромного количества элементов на одной форме интерфейса потребует от человека существенного напряжения, что приведет к усталости и ошибкам при работе.

Данные требования достаточно хорошо коррелируют с классическими показателями эффективности – результативностью, оперативностью и ресурсоэкономностью (в таком же порядке) [14]. Улучшение же выполнения каждого требования будет повышать соответствующий показатель.

Сложность повышения итоговой эффективности заключается в том, что, как правило, решения, применимые при проектировании интерфейсов, влияют на показатели с разным знаком. Для примера рассмотрим взаимное влияние двух из них: размещение всех элементов на первоначальной форме хотя и повысит оперативность, однако снизит ресурсоэкономность – пользовате-

лю будет сложно обработать такое количество информации сразу. А заведомое понижение оперативности разбиением полей ввода на несколько форм позволит работать с системой в более комфортном режиме (частным примером этого можно считать диалоговые интерфейсы [6], [7]) – пользователь на каждом шаге-форме будет концентрироваться на логически сгруппированных элементах. Аналогичные примеры могут быть найдены и для других пар показателей.

Способ сравнения эффективности взаимодействий. В большинстве случаев определение абсолютного значения эффективности Взаимодействия пользователя с Системой – не только крайне сложная, но и не обязательная задача. Как правило, практическая значимость оценки заключается в сравнении реализаций Сервисов, использующих данные Взаимодействия, для выбора наилучших из них или для улучшения существующего. При этом сравнение эффективностей двух принципиально разных Сервисов – т. е. относящихся к разным классам – может оказаться бессмысленным, поскольку решаемые в них задачи качественно различны.

Исходя из приведенных предпосылок, оценка может быть проведена с помощью последовательности шагов [15]:

Шаг 1. Выбрать класс сравниваемых Сервисов (и, соответственно, Взаимодействий, их интерфейсов и Систем).

Шаг 2. Построить модель Взаимодействий для класса таких Сервисов (далее – Модель).

Шаг 3. Определить алгоритмы вычисления каждого из показателей эффективности для выбранного класса Сервисов в терминах Модели.

Шаг 4. Вычислить для каждого Сервиса показатели эффективностей его Взаимодействия согласно осуществленной реализации последнего, т. е. его схемы (далее – Схема).

Шаг 5. Выбрать и применить метод сравнения по совокупности показателей – например, с помощью вычисления эффективности как корня из суммы квадратов показателей, назначением весов каждому показателю, применением ранжирования по Парето [16], [17] или даже простым выбором главного показателя для сравнения. На выбор также может повлиять область применения систем.

Развитием предложенного способа сравнения может стать решение обратной задачи – определе-

ние варианта реализации Взаимодействия для обеспечения требуемых показателей эффективности. Так, для одних сфер основным требованием будет высокая результативность (например, в банковской или военной сфере, где зачастую важна гарантия решения задачи), для других – высокая оперативность (для сферы безопасности – пожарной или информационной, где ущерб напрямую зависит от времени противодействия), для третьих – высокая ресурсоэкономность (для сферы обслуживания населения, которое, скорее всего, будет незнакомо с новыми интерфейсными решениями, что потребует повышения уровня комфортности при работе). В алгоритмическом виде способ может быть представлен с помощью следующего псевдокода:

```

Algorithm ComparisonOfEffectiveness()
Input:
    System_1 – Первый сервис для сравнения
    эффективности
    System_2 – Второй сервис для сравнения
    эффективности
    Area – Область применения сервисов
Output:
    Result – Результат сравнения (Error –
    ошибка, Success – ранжирование эффективностей
    взаимодействий)
Body:
    1: Begin
    // Шаг 1
    2: Class = SelectCommonClass(System_1,
    System_2)
    3: If (Class == None)
    4:   Return Error(“Сервисы относятся к
    разным классам”);
    // Шаг 2
    5: Model = SelectModel(Class);
    // Шаг 3
    6: Algorithms = SelectAlgorithms (Model);
    // Шаг 4
    7: Indicators_1 = CalculateIndicators(Model, Algorithms, System_1);
    8: Indicators_2 = CalculateIndicators(Model, Algorithms, System_2);
    // Шаг 5
    9: Comparator = SelectComparator(Area);
    10: Choice = PerformCompare(Comparator,
    Indicators_1, Indicators_2);
    11: If (Choice == First)
    12:   Return Success(“Взаимодействие в
    первом сервисе эффективнее, чем во втором”);
    13: Else
    14:   Return Success(“Взаимодействие во
    втором сервисе эффективнее, чем в первом”);
    15: End

```

В псевдокоде использованы следующие обозначения: *SelectCommonClass()* – функция выбора общего класса для двух Сервисов; *Class* – выбранный класс Сервиса; *Error()* – возвращение ошибки из алгоритма; *SelectModel()* – выбор Модели по заданному классу; *Model* – выбранная Модель; *SelectAlgorithm()* – выбор алгоритма по заданной Модели; *Algorithm* – выбранный алгоритм; *CalculateIndicators()* – вычисление показателей эффективности (результативность, оперативность, ресурсоэкономность) для Взаимодействия по заданным Модели и алгоритмам; *SelectComparator()* – выбор метода сравнения Взаимодействий по заданной области их применения их Сервисов; *Comparator* – выбранный метод сравнения; *PerformCompare()* – непосредственное сравнение двух Взаимодействий заданным методом и по заданным показателям эффективности; *Success()* – возвращение успешности сравнения с указанием более эффективного Взаимодействия.

Для подтверждения работоспособности и детализации особенностей способа проведем гипотетический эксперимент по сравнению эффективности различных Схем Взаимодействий для Сервисов одного класса.

Классы сервисов. Несмотря на огромное количество Сервисов (и их интерфейсов [18]), обеспечивающих Взаимодействие пользователя с информационной системой, можно выделить 3 достаточно распространенных класса, отличающихся акцентом на инициации процесса:

1. Запросный Сервис – предназначен для удовлетворения поисковых запросов пользователя, результатом чего становится некоторый объект (информация, предмет физического мира, действие и т. п.). Очевидно, что инициатором является пользователь, потребности которого стремится удовлетворить Сервис. Классическим примером можно считать интернет-поисковый сервис, который на основании введенных данных выдает исковую информацию (или ее множество). Примером, более относящимся к физическому миру, можно считать вендинговое оборудование.

2. Управляющий Сервис – предоставляет пользователю набор данных и требует от него принятия решения. В данном случае как раз сам Сервис иницирует Взаимодействие с пользователем, находящимся в режиме «покоя» (или безучастного наблюдения) [19]. В качестве примера можно привести стремительно развивающуюся область беспилотного транспорта, который, хотя

и функционирует без участия человека, однако может требовать от него ввода данных (указание конечной точки маршрута, контроль состояния здоровья пассажира и т. п.).

3. Игровой Сервис – промежуточный между запросным и управляющим, поскольку в нем пользователь и Сервис – равноправные «противники» информационного обмена (т. е. или могут инициировать Взаимодействие независимо, или же Взаимодействие происходит по заранее выбранным правилам). Типичным примером может стать игра в шахматы с компьютером, когда каждый делает ход согласно правилам игры, независимо от собственного «желания».

Далее будет проведен эксперимент по оценке и сравнению эффективностей Схем Взаимодействий (т. е. конкретных реализаций последних), относящихся к 1-му классу; поэтому рассмотрим его более детально. Несмотря на кажущуюся тривиальность класса, подобные системы могут быть достаточно сложными (например, в случае «витиеватых» Сценариев ввода данных, их проверки, частичной предварительной обработки Системой, многокаскадного поиска). Тем не менее упростим класс рассматриваемых Сервисов, приняв линейность их Сценариев. Так, общая логика их работы будет заключаться в такой последовательности действий: «Сбор данных пользователя → Обработка данных информационной системой → Поиск результата в базе данных → Возврат результата пользователю». Для построения Модели их Взаимодействия с пользователем учтем следующие особенности класса. Во-первых, исходя из того, что для осуществления запроса Системе требуется получить набор точно его специализирующих данных, основную логику работы должен составлять сбор таких данных у пользователя. Во-вторых, поскольку задача Системы состоит в выдаче результата запроса, выходные данные можно представить в виде одного блока информации (с возможностью последующей детализации на стороне пользователя). И, в-третьих, ориентированность подобных Сервисов на не всегда подготовленных пользователей должна позволять последним корректировать данные в случае неверного их ввода.

Обобщенно, любое Взаимодействие в Сервисах такого класса можно представить в виде набора полей для ввода данных, последовательно заполняемых пользователем, информация из которых передается в Систему для осуществления поиска. Для представления результата пользователю может использоваться лишь одно поле вывода.

Схемы взаимодействий для класса сервиса.

Прежде чем перейти к моделированию Взаимодействий данного класса Запросных Сервисов, применим научное предвидение касательно влияния Схем на показатели эффективности. Для этого рассмотрим следующие 3 реализации, различающиеся количеством использованных форм и плотностью их наполнения интерфейсными элементами (как среднее количество элементов на форме). Также, для упрощения понимания, возьмем количество элементов, посредством которых у пользователя запрашиваются данные, равное 4.

Выделим в Модели следующие аспекты: внешние – объекты интерфейса, состоящие из форм и элементов интерфейса (которые могут быть полями для ввода/вывода и кнопками перехода), а также внутренние – связи Сценария (которые определяют его логику) [20]. Таким образом, элементы интерфейса предназначены для следующих задач: ввод, вывод и переход. Для представления Модели будем использовать графическую нотацию, представленную на рис. 1 (подобно [21]).

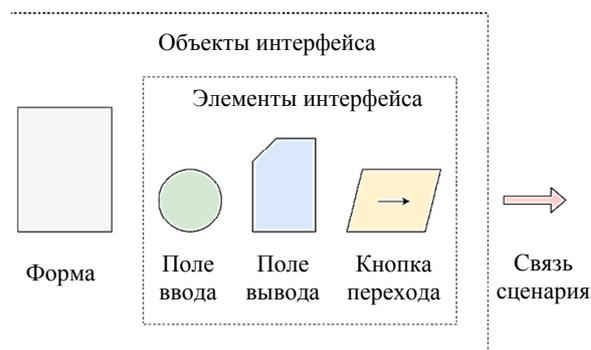


Рис. 2

Так, форма представляет собой область, используемую для размещения полей ввода/вывода и кнопок перехода на следующие (или предыдущие) формы. Связь Сценария же определяет последовательность переходов по точкам Сценария (в виде привязки к кнопкам форм, на которые они ведут).

При этом важно отметить, что все Сервисы решают одну задачу, определяемую их классом – запрашивают набор одинаковых данных у пользователя, передают его в Систему и возвращают единственный результат. И если конечные формы вывода результата у Сервисов будут одинаковы, то вводить данные можно разнообразными способами. Последние как раз и задают различные варианты реализации Взаимодействий – т. е. Схемы; они будут рассмотрены далее.

Схема при высокой плотности элементов (СВПЭ). Основное отличие данной Схемы от остальных состоит в том, что все элементы ввода расположены на одной (первой форме), а вторая (последняя) используется для вывода. Таким образом достигается максимально высокая плотность элементов. К основным дефектам Взаимодействий для данной Схемы, описанным в предыдущих частях цикла [6], [7], могут быть отнесены следующие:

– «Дефект_8. Переполнение интерфейсных форм элементами», поскольку это является отличительной особенностью данной высокоплотной Схемы;

– «Дефект_11. Игнорирование ограничений организма человека», частично присутствующий по причине отсутствия группировки и невозможности охвата пользователем всего множества отображаемых полей на одной форме.

Обобщенный интерфейс Взаимодействия в запросном Сервисе для данной Схемы может быть представлен рис. 2.

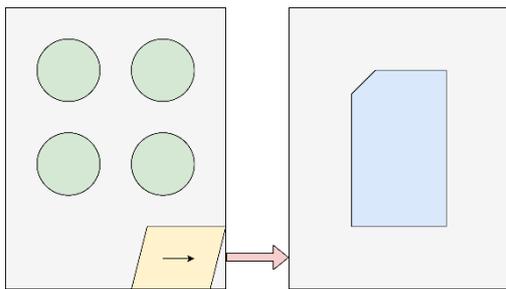


Рис. 2

Согласно рисунку, первая форма содержит 4 поля ввода и кнопку перехода к следующей форме; вторая же форма содержит только поле вывода. Визуальный анализ позволяет сделать следующие предположения касательно показателей эффективности. Во-первых, результативность такого

Взаимодействия должна быть высокой, поскольку пользователь всегда сможет перейти на форму с выводом результатов запроса. Во-вторых, оперативность будет высокой, поскольку ему придется сделать практически минимальное количество действий – ввести данные, нажать кнопку перехода и получить результат (помимо банальных переходов между объектами интерфейса). Однако ресурсоэкономность должна быть достаточно низкой, поскольку отображение на одной форме большого количества элементов будет иметь высокую (по крайней мере выше, чем минимальная) когнитивную нагрузку на пользователя.

Схема при низкой плотности элементов (СНПЭ). В противоположность СВПЭ, низкая плотность элементов означает нахождение на форме минимально возможного их количества, т. е. по одному полю ввода и минимально необходимому числу кнопок. Как следствие, возрастает количество форм, что ведет к добавлению кнопок перехода на предыдущие формы (для корректировки пользователем неверно введенных данных). К основным дефектам Взаимодействий для данной Схемы, описанным в [6], [7], могут быть отнесены следующие:

– «Дефект_4. Цикличность пути сценария», поскольку в Сценарии присутствуют переходы на предыдущие формы;

– «Дефект_5. Сложность путей сценария» по причине большого количества переходов между формами.

Обобщенный интерфейс Взаимодействия в запросном Сервисе для данной Схемы может быть представлен рис. 3.

Согласно рисунку, интерфейс состоит из 4 форм, содержащих по 1 полю ввода данных, и форму с полем вывода. Также каждая форма для

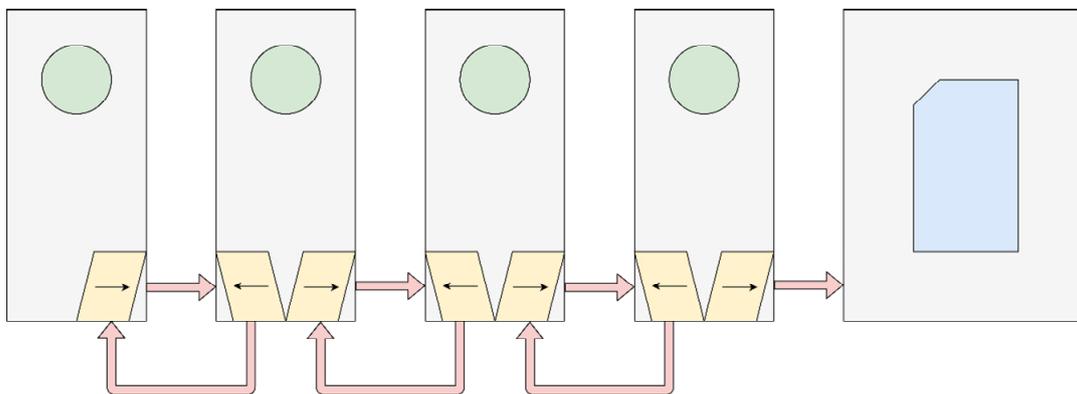


Рис. 3

ввода содержит кнопки перехода на следующую, а каждая, кроме первой, – кнопку перехода на предыдущую форму. Результаты аналогичного визуального анализа следующие. Во-первых, результативность такого Взаимодействия должна быть низкой, поскольку пользователь на большинстве форм может перейти на предыдущую, отклоняясь тем самым от попадания на форму с результатом запроса – присутствует множество Сценарных разветвлений. Во-вторых, оперативность также будет низкой, поскольку помимо действий по вводу данных появятся дополнительные – нажатия на кнопки перехода на новых (по сравнению с СВПЭ) формах. Однако ресурсоэкономность должна резко возрасти, поскольку при работе с каждой формой пользователь будет получать практически минимальное количество визуальной информации для обработки – одно поле ввода и одна или две кнопки перехода (помимо последней формы с единственным полем вывода).

Схема при средней плотности элементов (ССПЭ). Данную Схему можно считать промежуточной между СВПЭ и СНПЭ, поскольку в ней достигается некоторый баланс между формами и элементами. Добиться этого можно, выделяя равное количество форм и полей ввода, – таким образом, их количество будет равно \sqrt{N} везде, где N – число требуемых для ввода данных; как и ранее, завершать Сценарий Взаимодействия будет форма с полем вывода результата запроса.

Обобщенный интерфейс Взаимодействия в запросном Сервисе для данной Схемы может быть представлен рис. 4.

Согласно рисунку, интерфейс состоит из 2 форм, содержащих по 2 поля ввода данных (помимо кнопок перехода), и стандартную форму с

полем вывода. Результаты визуального анализа позволяют говорить о средних значениях показателей эффективности по следующим причинам: количество Сценарных ветвлений представляется разумным (поскольку поля ввода на каждой форме сгруппированы), необходимые для пользователя действия увеличены лишь переходами между группами элементов, а на каждой форме пользователь видит небольшое количество полей ввода, соответствующее количеству групп.

Также следующие дефекты частично присутствуют во всех приведенных Схемах:

– «Дефект_12. Отсутствие ответной реакции на действия» и «Дефект_13. Слабая адаптируемость под пользователя» по причине заведомо принятой простоты интерфейса без механизма подтверждения;

– «Дефект_17. Нехватка помощи пользователю» по причине той же простоты интерфейса и отсутствия подсказок к элементам и действиям;

– «Дефект_18. Возможность скрытых информационных воздействий через элементы», поскольку никак не контролируется механизм создания данных с результатом запроса, потенциально содержащих вредоносную информацию, воздействующую на пользователя.

Модель взаимодействия для класса сервиса. Одна из Моделей (подобных и построенных на принципах, аналогичных [22]–[25]), описывающих Взаимодействие в подобных классах Сервисов (упрощенную, как было указано ранее), может представлять собой набор входных форм с полями ввода данных и выходной формы с полем вывода результата запроса при последовательном переключении от первой формы к последней по нажатию клавиши (естественно, при возможно-

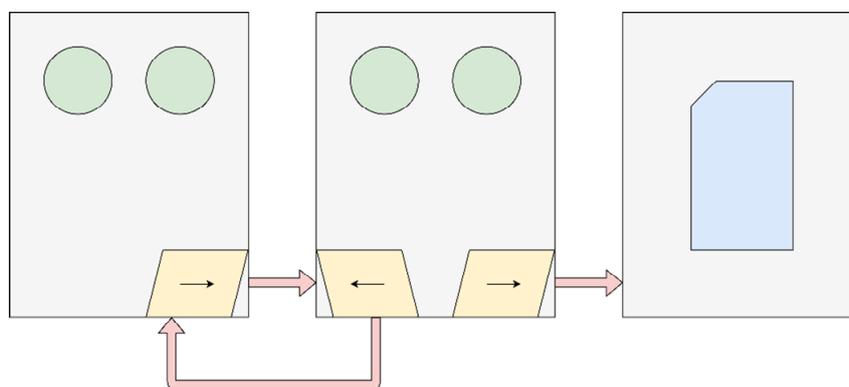


Рис. 4

сти «отката» к предыдущей форме для коррекции ошибочного ввода); это было в достаточной степени продемонстрировано на различных Схемах (см. рис. 2–4). В аналитическом виде модель M может быть описана с помощью следующих двух частей.

Первая часть Модели вводит ее основной базис:

$$\left\{ \begin{array}{l} M = \langle F, E, B, T \rangle; \\ F = \left(\bigcup_{i=1}^{NF} F_i^{\text{In}}, F^{\text{Out}} \right); \\ E = \left(\bigcup_{j=1}^{NE} E_j^{\text{In}}, E^{\text{Out}} \right); \\ B = \left(\bigcup_{k=1}^{NF} B_k^{\text{Next}}, \bigcup_{l=1}^{NF-1} B_l^{\text{Prev}} \right); \\ T = \left(\bigcup_{m=1}^{NF} T_m^{\text{Next}}, \bigcup_{n=1}^{NF-1} T_n^{\text{Prev}} \right), \end{array} \right.$$

где F – множество форм (сокр. от *англ.* Form), каждая из которых используется для ввода запроса (F_i^{In} в числе NF) и вывода результата (F^{Out} в единичном числе); NF – число всех форм, используемых для ввода данных; E – множество элементов-полей (сокр. от *англ.* Element), каждый из которых используется для ввода одного данного (E_j^{In} в числе NE) и вывода одного результата (E^{Out} в единичном числе); NE – число всех элементов-полей, используемых для ввода данных; B – множество элементов-кнопок (сокр. от *англ.* Button), каждая из которых используется для перехода на следующую (B_k^{Next} в числе NF) или предыдущую (B_l^{Prev} в числе, на единицу меньшем NF) форму; T – множество Сценарных переходов (сокр. от *англ.* Transition), каждый из которых используется для связи кнопок переходов и целевых форм (T_m^{Next} в числе NF для переходов вперед и T_n^{Prev} в числе $NF - 1$ для перехода назад). В данном случае Сценарный переход «вперед» соответствует продвижению к точке получения Системой запроса для обработки, а Сценарный переход «назад» – в обратную сторону соответственно.

Вторая часть Модели устанавливает связи между ее базисом (числа в конце строк с уравнениями использованы для их нумерации):

$$\left. \begin{array}{l} (1) \quad \forall j(1 \leq j \leq NE), \exists i(1 \leq i \leq NF), E_j^{\text{In}} \in F_i^{\text{In}}; \\ (2) \quad \nexists i(1 \leq i \leq NF), \emptyset_E \in F_i^{\text{In}}; \\ (3) \quad E^{\text{Out}} \in F^{\text{Out}}; \\ (4) \quad \forall k(1 \leq k \leq NF), B_k^{\text{Next}} \in F_k^{\text{In}}; \\ (5) \quad \left\{ \begin{array}{l} \nexists k(1 \leq k \leq NF), \emptyset_B \in F_k^{\text{In}}; \\ \forall l(1 \leq l \leq NF - 1), B_l^{\text{Prev}} \in F_{l+1}^{\text{In}}; \end{array} \right. \\ (6) \quad \forall m(1 \leq m \leq NF - 1), T_m^{\text{Next}} \equiv \langle B_m^{\text{In}}, F_{m+1}^{\text{In}} \rangle; \\ (7) \quad T_{NF}^{\text{Next}} \equiv \langle B_{NF}^{\text{In}}, F^{\text{Out}} \rangle; \\ (8) \quad T_{NF}^{\text{Next}} \equiv \langle B_{NF}^{\text{In}}, F^{\text{Out}} \rangle; \\ (9) \quad \forall n(1 \leq n \leq NF - 1), T_n^{\text{Prev}} \equiv \langle B_{n+1}^{\text{In}}, F_n^{\text{In}} \rangle, \end{array} \right\}$$

где каждая строка системы уравнений имеет следующий смысл:

1. Любое поле ввода данных принадлежит одной из форм ввода запроса.
2. Не существует формы ввода запроса, которой не принадлежит ни одного поля ввода данных (\emptyset_E соответствует пустому множеству элементов).
3. Поле вывода результата принадлежит соответствующей форме (последней в Сценарии).
4. Любая кнопка для перехода на следующую форму принадлежит одной из форм для ввода запроса.
5. Не существует формы ввода запроса, которой не принадлежит ни одна кнопка перехода на следующую форму (\emptyset_B соответствует пустому множеству кнопок).
6. Любая кнопка для перехода на предыдущую форму принадлежит одной из форм для ввода запроса, кроме первой.
7. Любой переход вперед по Сценарию, кроме последнего, определяется кортежем кнопки перехода на следующую форму ввода запроса и этой формой.
8. Последний переход вперед по Сценарию определяется кортежем кнопки перехода на форму вывода результата запроса и этой формой.
9. Любой переход назад по Сценарию определяется кортежем кнопки перехода на предыдущую форму и этой формой.

Естественно, данная Модель описывает все множество Взаимодействий, даже когда на одной форме ввода запроса расположены сразу несколько кнопок перехода на следующую форму. Тем не менее, такой Сервис имеет право на существование, хотя и некорректно спроектирован изначально.

Расчет показателей эффективности. Предложим формулы расчета показателей эффективности для класса запросных Сервисов в терминах введенной Модели. Поскольку абсолютные значения эффективности отдельного Взаимодействия имеют мало практического смысла, а важна лишь возможность сравнения эффективностей нескольких Взаимодействий, то можно в качестве значения каждого показателя взять степень достижения некоторого идеального – т. е. для Сервиса, обладающей максимальной эффективностью Взаимодействия. Последнее будет представлять собой одну форму с одним полем ввода, используемым для ввода и для вывода и занимающим всю площадь формы, будучи при этом кнопкой для отправки запроса. По аналогии с предыдущими, обобщенный интерфейс идеального Взаимодействия для запросного Сервиса может быть представлен рис. 5.

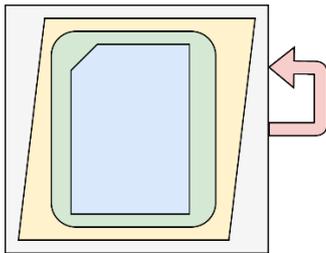


Рис. 5

Очевидно, что данный Сервис вряд ли будет существовать в реальности, однако он выступает как некая абстракция, относительно которой будут оцениваться остальные. Для идеальных показателей разумно выбрать нормированное значение 1.

Результативность, согласно введенному понятию, должна характеризовать гарантированность достижения пользователем момента отправки запроса в Систему. Факторами, негативно влияющими на показатель, будут всевозможные Сценарные ответвления (даже если они оправданы). Тогда значение показателя результативности (P от *англ.* Potency) будет зависеть от доли Сценарных переходов вперед среди всех Сценарных переходов:

$$P = \frac{\text{NUM}(T_m^{\text{Next}})}{\text{NUM}(T_m^{\text{Next}}) + \text{NUM}(T_m^{\text{Prev}})},$$

где оператор NUM() соответствует определению количества элементов множества. Так, если в интерфейсе отсутствуют Сценарные переходы

назад, то $P = \frac{1}{1} = 1$, что соответствует идеальному Взаимодействию. Если же в интерфейсе бесконечное количество переходов назад, то $P = \frac{1}{\infty} = 0$, т. е. пользователь гарантированно не дойдет до отправки запроса.

Используя Модель, можно получить окончательную формулу для вычисления показателя результативности:

$$P = \frac{NF}{NF + (NF - 1)} = \frac{NF}{2NF - 1}, \quad (1)$$

поскольку каждая форма содержит Сценарный переход вперед (в количестве NF), а каждая форма, кроме первой, – Сценарный переход назад (в количестве $NF - 1$).

Оперативность характеризует длительность достижения пользователем момента отправки запроса в Систему. Точная оценка длительности в единицах времени (например, в секундах) имеет существенные сложности по причине различной скорости работы пользователей с объектами интерфейса и зависимости от размера вводимых данных. Поэтому введем следующее упрощение, которое существенно не повлияет на результаты сравнения показателей двух Взаимодействий. Будем измерять длительность в атомарных действиях пользователя при работе с интерфейсом; для Модели выбранного класса Сервиса действия будут состоять из перемещения курсора к форме, выбора элемента на форме, ввода данных в элемент, перемещения к кнопке, нажатия на кнопку. Само же значение оперативности (O от *англ.* Operativeness) будет вычисляться как обратное значение минимальной длительности в атомарных действиях:

$$O = \frac{1}{\text{NUM}[\text{MIN}(\text{Action}_p)]},$$

где Action_p – атомарное действие; MIN() – минимальный набор атомарных действий до отправки запроса. Очевидно, что оперативность для идеального Взаимодействия будет тождественна 1, поскольку в ней потребуется лишь одно действие – ввод данных, который также инициирует и их отправку.

Используя Модель, можно получить окончательную формулу для вычисления показателя оперативности:

$$O = \frac{1}{NF + NE + NE + NF + NF} = \frac{1}{3NF + 2NE}, \quad (2)$$

поскольку атомарные действия состоят из следующих (см. слагаемые знаменателя формулы): переход на каждую форму (в количестве NF), переход на каждый элемент всех форм (в количестве NE) и ввод данных через него (в количестве NE), переход на кнопку перехода на следующую форму (в количестве NF) и нажатие на нее (в количестве NF).

Ресурсоэкономность (обратная ресурсоемкости) определяет когнитивную нагрузку на пользователя и может быть сопоставлена со средним количеством объектов на каждом Сценарном шаге (включая саму форму, поля для ввода/вывода и кнопки). Таким образом, для вычисления показателя достаточно посчитать все элементы на формах и усреднить их по количеству форм. Значение показателя (R от *англ.* Resource) будет вычисляться следующим образом:

$$R = \frac{1}{\frac{\text{NUM}(\text{OBJECTS}(F_i^{\text{In}})) + \text{NUM}(\text{OBJECTS}(F^{\text{Out}}))}{\text{NUM}(F_i^{\text{In}}) + \text{NUM}(F^{\text{Out}})}}$$

где $\text{OBJECTS}()$ – число объектов в момент отображения указанной формы на Сценарном шаге.

Очевидно, что ресурсоэкономность для идеального Взаимодействия должна быть тождественна 1, поскольку она состоит из 1 объекта, включающего в себя все возможные объекты интерфейса – формы и элементы (поля ввода/вывода, кнопки).

Используя Модель, можно получить окончательную формулу для вычисления показателя оперативности:

$$R = \frac{1}{\frac{(NF + 1) + NE + 1 + NF + (NF - 1)}{NF + 1}} = \frac{NF + 1}{3 \times NF + NE + 1}, \quad (3)$$

поскольку число всех форм (включая последнюю – для вывода) равно $NF + 1$, а число всех объектов на всех Сценарных шагах состоит из самой формы (в количестве $NF + 1$), полей ввода (в количестве NE), поля вывода (в количестве 1), кнопки перехода вперед на каждой форме, кроме

последней (в количестве NF), кнопки перехода назад на каждой форме, кроме первой и последней (в количестве $NF - 1$).

Эффективность Взаимодействия как совокупность 3 ее показателей может быть записана следующим образом:

$$E = \langle P, O, R \rangle,$$

т. е. как кортеж независимых значений. Сравнение же Взаимодействий по их эффективностям в этом случае может быть сделано при помощи ранжирования по Парето, суть которого заключается в разделении сравниваемых по критериям объектов так, чтобы объекты высшего ранга доминировали бы по всем критериям над объектами низшего ранга. Однако в ряде случаев достаточным будет использование среднеквадратичной эффективности как расстояния в пространстве ее показателей:

$$\overline{E_{POR}} = \sqrt{\frac{P^2 + O^2 + R^2}{3}}.$$

Заметим, что для идеального Взаимодействия, у которого все показатели равны 1, модуль эффективности также равен 1, что подчеркивает корректность выбранной идеализированной абстракции.

Как будет показано далее, интересны с научной и практической точки зрения могут быть средние значения эффективности для пар показателей (например, если третьим можно пренебречь):

$$\begin{cases} \overline{E_{PO}} = \sqrt{\frac{P^2 + O^2}{2}}; \\ \overline{E_{PR}} = \sqrt{\frac{P^2 + R^2}{2}}; \\ \overline{E_{OR}} = \sqrt{\frac{O^2 + R^2}{2}}. \end{cases}$$

Эксперимент. Следующим шагом для сравнения Взаимодействий должно стать вычисление их эффективности через ее показатели согласно полученным формулам (1)–(3). Проведем оценку трех предложенных Схем в общем случае, моделируя их для целого диапазона количеств элементов-полей ввода данных: $1 \leq NE \leq 10$. Как будет показано далее, верхняя граница диапазона достаточна, количество же форм определяется числом NF , которое или постоянно (для СВПЭ), или зависит от NE (для СНПЭ и ССПЭ).

Таблица 1

NE	P	O	R	\bar{E}	$\overline{E_{PO}}$	$\overline{E_{PR}}$	$\overline{E_{OR}}$
1	1	0.2000	0.4000	0.6325	0.7211	0.7616	0.3162
2	1	0.1429	0.3333	0.6141	0.2564	0.7454	0.2564
3	1	0.1111	0.2857	0.6039	0.2168	0.7354	0.2168
4	1	0.0909	0.2500	0.5974	0.1881	0.7289	0.1881
5	1	0.0769	0.2222	0.5931	0.1663	0.7244	0.1663
6	1	0.0667	0.2000	0.5900	0.1491	0.7211	0.1491
7	1	0.0588	0.1818	0.5878	0.1351	0.7187	0.1351
8	1	0.0526	0.1667	0.5861	0.1236	0.7169	0.1236
9	1	0.0476	0.1538	0.5848	0.1139	0.7154	0.1139
10	1	0.0435	0.1429	0.5838	0.1056	0.7143	0.1056

Таблица 2

NE	P	O	R	\bar{E}	$\overline{E_{PO}}$	$\overline{E_{PR}}$	$\overline{E_{OR}}$
1	1.0000	0.2000	0.4000	0.6325	0.7211	0.7616	0.3162
2	0.6667	0.1000	0.3333	0.4342	0.2461	0.5270	0.2461
3	0.6000	0.0667	0.3077	0.3912	0.2226	0.4768	0.2226
4	0.5714	0.0500	0.2941	0.3722	0.2110	0.4544	0.2110
5	0.5556	0.0400	0.2857	0.3614	0.2040	0.4417	0.2040
6	0.5455	0.0333	0.2800	0.3545	0.1994	0.4335	0.1994
7	0.5385	0.0286	0.2759	0.3497	0.1961	0.4278	0.1961
8	0.5333	0.0250	0.2727	0.3461	0.1937	0.4236	0.1937
9	0.5294	0.0222	0.2703	0.3434	0.1918	0.4203	0.1918
10	0.5263	0.0200	0.2683	0.3413	0.1902	0.4177	0.1902

Расчет для СВПЭ. Для данной Схемы $NF = 1$ и, следовательно, формулы для вычисления показателей эффективности имеют следующий вид:

$$\begin{cases} P = 1; \\ O = \frac{1}{2 \times NE + 3}; \\ R = \frac{2}{NE + 4}. \end{cases}$$

Расчет показателей эффективности и вариантов ее среднеквадратичного значения для схем при высокой плотности элементов представлен в табл. 1.

Расчет для СНПЭ. Для данной Схемы $NF = NE$ и, следовательно, формулы для вычисления показателей эффективности имеют следующий вид:

$$\begin{cases} P = \frac{NE}{2 \times NE - 1}; \\ O = \frac{1}{5 \times NE}; \\ R = \frac{NE + 1}{4 \times NE + 1}. \end{cases}$$

Расчет показателей эффективности и ее среднеквадратичное значение для схем при низкой плотности элементов представлен в табл. 2.

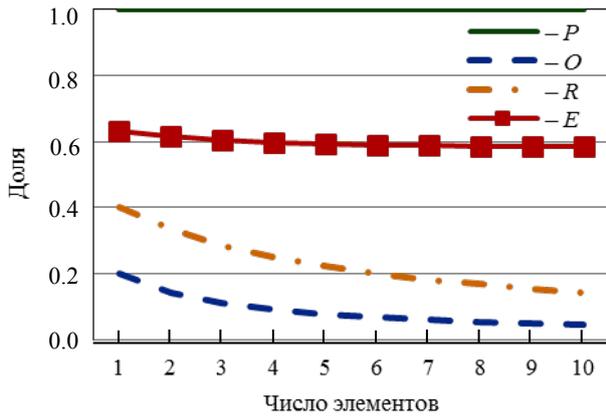
Расчет для ССПЭ. Для данной Схемы $NF = \sqrt{NE}$ и, следовательно, формулы для вычисления показателей эффективности имеют следующий вид:

$$\begin{cases} P = \frac{\sqrt{NE}}{2 \times \sqrt{NE} - 1}; \\ O = \frac{1}{3 \times \sqrt{NE} + 2 \times NE}; \\ R = \frac{\sqrt{NE} + 1}{3 \times \sqrt{NE} + NE + 1}. \end{cases}$$

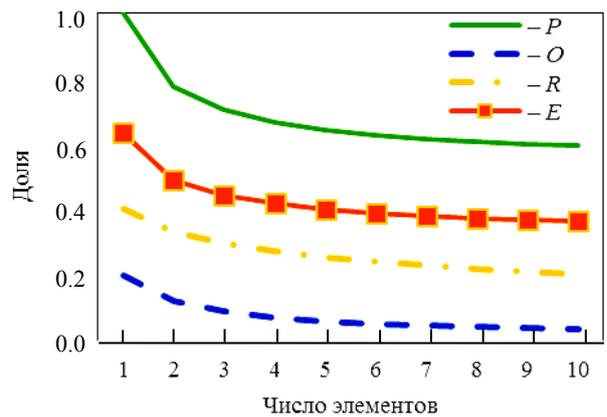
Расчет показателей эффективности и ее среднеквадратичное значение для схем при средней плотности элементов представлен в табл. 3.

Таблица 3

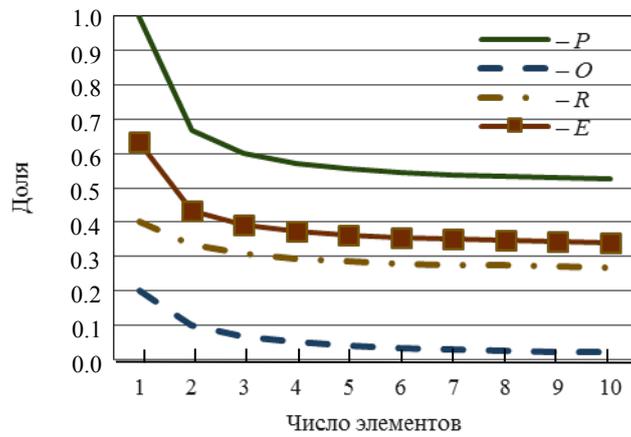
NE	P	O	R	\bar{E}	$\overline{E_{PO}}$	$\overline{E_{PR}}$	$\overline{E_{OR}}$
1	1.0000	0.2000	0.4000	0.6325	0.7211	0.7616	0.3162
2	0.7735	0.1213	0.3333	0.4913	0.2508	0.5955	0.2508
3	0.7029	0.0893	0.2971	0.4436	0.2194	0.5396	0.2194
4	0.6667	0.0714	0.2727	0.4179	0.1994	0.5093	0.1994
5	0.6440	0.0599	0.2546	0.4013	0.1850	0.4897	0.1850
6	0.6282	0.0517	0.2404	0.3895	0.1739	0.4756	0.1739
7	0.6165	0.0456	0.2288	0.3806	0.1649	0.4650	0.1649
8	0.6074	0.0408	0.2190	0.3735	0.1575	0.4565	0.1575
9	0.6000	0.0370	0.2105	0.3677	0.1512	0.4496	0.1512
10	0.5939	0.0339	0.2032	0.3629	0.1456	0.4438	0.1456



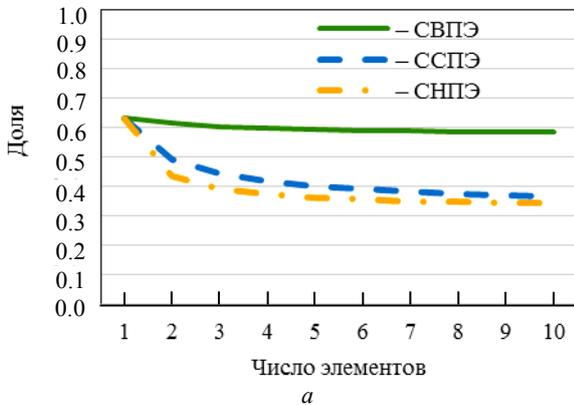
a



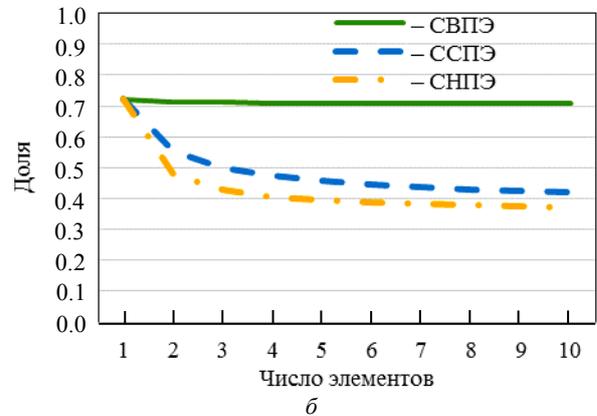
б



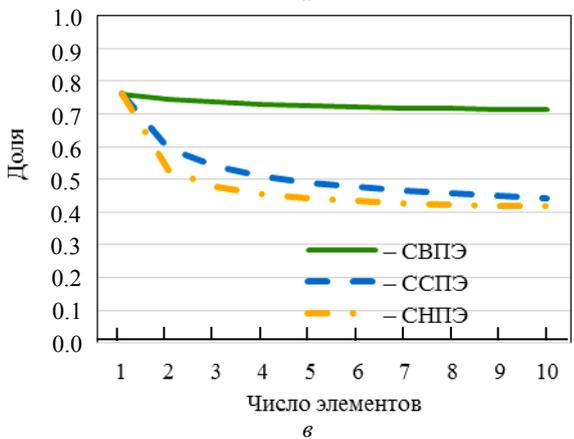
в
Рис. 6



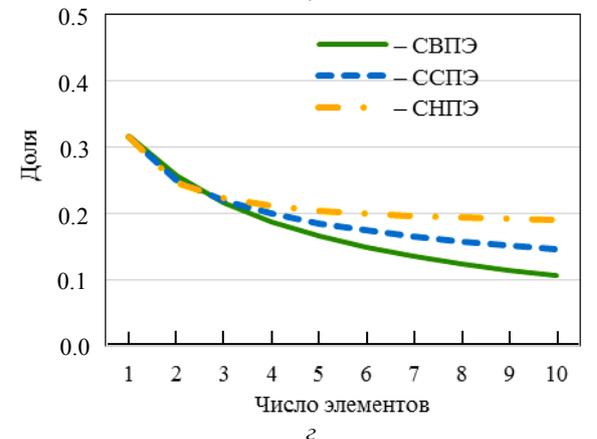
а



б



в



г

Рис. 7

В графическом виде расчет показателей эффективности и ее среднеквадратичного значения представлены на рис. 6: a – для СВПЭ; b – для ССПЭ; v – для СНПЭ.

Графики сравнения эффективности Взаимодействий для Схем представлен на рис. 7 следующим образом: a – по всем показателям; b – по результативности и оперативности; v – по результативности и ресурсоэкономности; z – по оперативности и ресурсоэкономности.

Применим для сравнения эффективностей ранжирование по Парето. Поскольку представление показателей в 3D-мерном пространстве затруднительно для визуального анализа, проведем ранжирование каждой плоскости пространства (т. е. для пар показателей). Метод ранжирования может быть представлен с помощью площадей под точками на плоскости. Точки, площади которых пересекаются, относятся к одному рангу, а точка с площадью, полностью входящей в площадь другой точки, относится к меньшему рангу. Интерпретируем графически метод для точки при $NE = 5$, поскольку результаты будут одинаковыми для любого количества элементов; на рис. 8 она представлена следующим образом: a – для плоскости результативности и оперативности; b – для плоскости результативности и ресурсоэкономности; v – для плоскости оперативности и ресурсоэкономности.

Согласно рис. 8, a для плоскости результативности и оперативности площадь под точкой для СНПЭ входит в площадь ССПЭ, которая в свою очередь входит в площадь СВПЭ. Для остальных плоскостей (рис. 8, b и v) площади под точками взаимно пересекаются.

Анализ результатов. Проанализируем результаты проведенного эксперимента по моделированию различных Схем Взаимодействий для запросных Сервисов при различных значениях количества элементов.

Во-первых, динамика показателей эффективности всех схем (см. рис. 6) показывает постепенное уменьшение их значений, а ранжирование показателей друг относительно друга определяется первым значением (при $NE = 1$) и не меняется со временем. Такая ситуация объясняется рядом внесенных упрощений, простотой самого класса запросных Сервисов и определенной линейностью выбранной для него модели.

Во-вторых, СВПЭ обладает большей среднеквадратичной эффективностью (рис. 7, a) вследствие ее высокой результативности и таким же весом, как и у всех остальных показателей. Последнее представляется не всегда корректным, поскольку в большинстве случаев пользователь все же дойдет до Сценарной точки отправки запроса в Систему и более важными показателями для него будут оперативность (т. е. как быстро он отправит запрос и получит ответ) и ресурсоэкономность (т. е. как сильно он «устанет» при этом). За данной Схемой следует ССПЭ, а затем и СНПЭ.

В-третьих, участие результативности в расчете эффективности по парам показателей приводит к такой же картине (рис. 7, b и v), как и при расчете эффективности по всем показателям, притом с большим отрывом СВПЭ.

В-четвертых, подсчет эффективностей по показателям оперативности и ресурсоэкономности (рис. 7 z) при $NE \geq 3$ приводит к выходу на первое место СНПЭ, за которой следует ССПЭ, а затем и СВПЭ (т. е. ситуация, обратная графику среднеквадратичной эффективности). Таким образом, без учета показателя результативности, по которому СНПЭ очевидно проигрывает (по причине множества вариантов отклонения от движения вперед по Сценарному пути), здесь она начинает выигрывать.

И, в-пятых, применение ранжирования по Парето показало, что оно дает результат лишь для пары результативность–оперативность – для них больший ранг имеет СВПЭ, затем ССПЭ и СНПЭ. Для других пар показателей все Схемы могут быть отнесены к одному рангу. Как следствие, ранжирование всех трех показателей будет безрезультатным, что означает одинаковость рангов всех схем.

Как само понятие эффективности взаимодействия пользователя с информационной системой, так и ее сравнение для различных реализаций – это сложнейшая задача. В статье было предложено решение, заключающееся в делении всех информационных сервисов (сочетающих в себе информационную систему и взаимодействие с ней пользователя) на классы, обладающие собственными моделями. В терминах последних были предложены способы расчета показателей эффективности. В результате появился математический

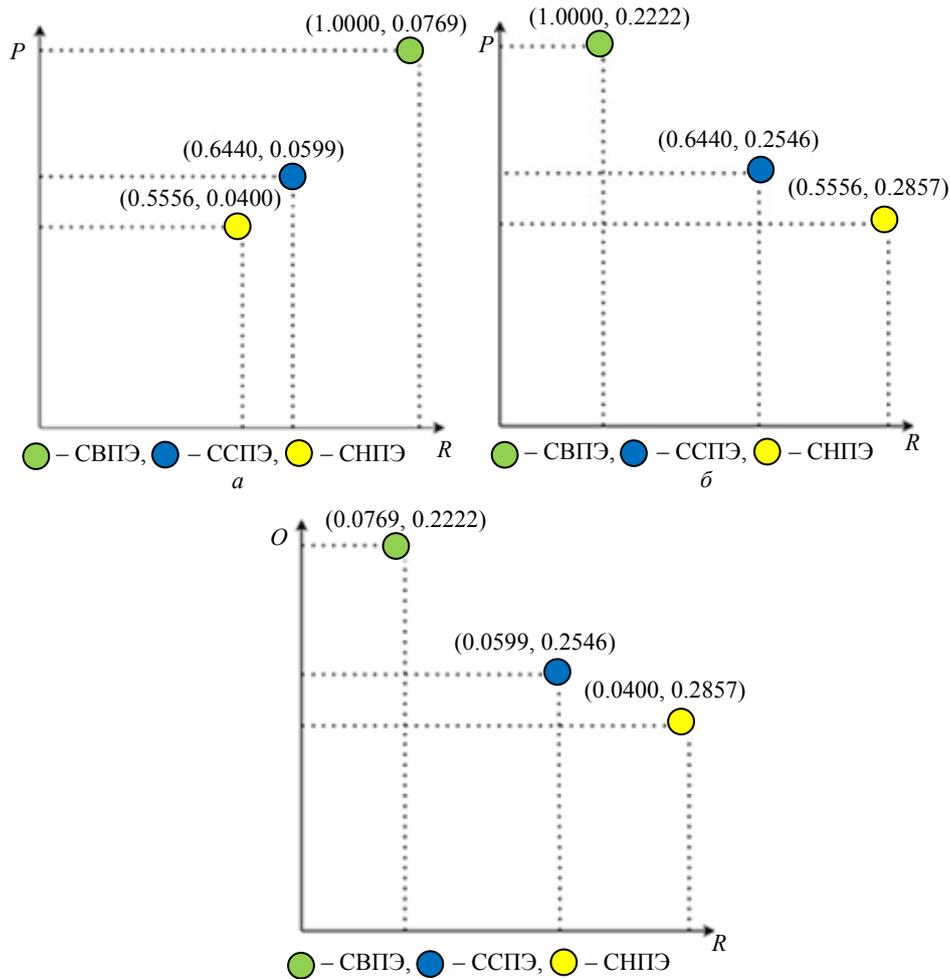


Рис. 8

аппарат для численного сравнения взаимодействий. Апробация аппарата на трех реализациях взаимодействий показала его работоспособность и общую корректность получаемых результатов.

Продолжением исследования должно стать развитие решения обратной задачи – по выбору внешнего вида интерфейса и внутренней логики

взаимодействия его объектов согласно заданным условиям на показатели эффективности. Таким образом, получаемое взаимодействие станет адаптивным, поскольку будет динамически подстраиваться под потребности и возможности пользователя.

СПИСОК ЛИТЕРАТУРЫ

1. Визуализация информационного обмена в условиях концепции Industry 4.0 / Ю. Н. Волщук, М. Ю. Волщук, К. Е. Израилов, А. В. Романенко // Автоматизация в промышленности. 2020. № 8. С. 23–29.

2. Покусов В. В. Оценка эффективности системы обеспечения ИБ. Ч. 1. Показатели и модели представления // Защита информации. Инсайд. 2019. № 2 (86). С. 54–60.

3. Покусов В. В. Оценка эффективности системы обеспечения ИБ. Ч. 2. Методика и результаты // Защита информации. Инсайд. 2019. № 3 (87). С. 64–72.

4. Основные принципы проектирования архитектуры современных систем защиты / М. В. Буйне-

вич, К. Е. Израилов, В. В. Покусов, А. Ю. Ярошенко // Национальная безопасность и стратегическое планирование. 2020. № 3 (31). С. 51–58.

5. Яковлев А. В., Васюкова Е. О., Пеливан М. А. Моделирование взаимодействия клиента с сервером в информационной системе // Евразийский союз ученых. 2015. № 10-2 (19). С. 184–187.

6. Курта П. А. Взаимодействие пользователя с информационной системой. Ч. 2. Алгоритмы обнаружения недостатков // Изв. СПбГЭТУ ЛЭТИ. 2020. № 10. С. 34–44.

7. Курта П. А. Взаимодействие пользователя с информационной системой. Ч. 1. Схема взаимодей-

ствия и классификация недостатков // Изв. СПбГЭТУ ЛЭТИ. 2020. № 8–9. С. 35–45.

8. Радванская Л. Н., Удовиченко Л. В., Ходаков Д. В. Оценка эффективности интерфейса «пользователь – компьютеризированная система» // Радиоэлектроника и информатика. 1999. № 3 (8). С. 101–103.

9. Яковлев Ю. С., Курзанцева Л. И. Модель учебной смарт-системы пользователя и оценка ее реализации // Математические машины и системы. 2019. № 3. С. 86–100.

10. Корвяков В. П. Метод нейро-нечеткой оценки пригодности использования графического интерфейса пользователя // Вестн. Моск. гос. тех. ун-та им. Н. Э. Баумана: Сер. «Приборостроение». 2016. № 5 (110). С. 61–74.

11. Ивкин А. С., Анохин А. Н., Чепурко В. А. Подход к эмпирической оценке человеко-машинного интерфейса для управления технологическим процессом: на примере экологического интерфейса для управления уровнем воды в барабане-сепараторе АЭС типа РБМК // Биотехносфера. 2015. № 1 (37). С. 26–37.

12. Ахунова Д. Г., Вострых А. В., Курта П. А. Оценка пользовательского интерфейса информационных систем посредством моделей качества программного обеспечения // Информатизация и связь. 2020. № 2. С. 127–135.

13. Израилов К. Е. Система критериев оценки способов поиска уязвимостей и метрика понятности представления программного кода // Информатизация и связь. 2017. № 3. С. 111–118.

14. Курта П. А. Подход к оценке эффективности интерфейсов программ с использованием классической базы // Пожарная безопасность: современные вызовы. Проблемы и пути решения: материалы Междунар. науч.-практ. конф. Санкт-Петербург: Санкт-Петербург. ун-т Гос. противопожарной службы МЧС России. 2020. С. 88–90.

15. 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.

16. Буйневич М. В., Ахунова Д. Г., Ярошенко А. Ю. Комплексный метод решения типовой задачи риск-менеджмента в инфологической среде (на примере ранжирования требований пожарной безопасности). Ч. 1 // Науч.-аналит. журн. «Вестн. Санкт-Петерб. университета Гос. противопожарной службы МЧС России». 2020. № 3. С. 88–99.

17. Буйневич М. В., Ахунова Д. Г., Ярошенко А. Ю. Комплексный метод решения типовой задачи риск-менеджмента в инфологической среде (на примере ранжирования требований пожарной безопасности). Ч. 1 // Науч.-аналит. журн. «Вестн. С.-Петерб. ун-та Гос. противопожарной службы МЧС России». 2020. № 4. С. 78–89.

18. Курта П. А., Коломеец М. В. Обобщенная классификация интерфейсов транспортной инфраструктуры «Умного города» // Вестн. кибернетики. 2020. № 4 (40). С. 6–13.

19. Израилов К. Е., Обрезков А. И., Курта П. А. Подход к выявлению последовательности одноцелевых сетевых атак с визуализацией их прогресса эксперту // Методы и технические средства обеспечения безопасности информации. 2020. № 29. С. 68–69.

20. Буйневич М. В., Израилов К. Е., Щербаков О. В. Структурная модель машинного кода, специализированная для поиска уязвимостей в программном обеспечении автоматизированных систем управления // Проблемы управления рисками в техносфере. 2014. № 3 (31). С. 68–74.

21. Буйневич М. В., Покусов В. В., Израилов К. Е. Способ визуализации модулей системы обеспечения информационной безопасности // Науч.-аналит. журн. «Вестн. Санкт-Петербургского ун-та Гос. противопожарной службы МЧС России». 2018. № 3. С. 81–91.

22. Волобуев С. В., Волобуев Е. С. Моделирование информационного взаимодействия систем // Вопр. защиты информации. 2003. № 3 (62). С. 54–61.

23. Математическое моделирование взаимодействия исследуемой системы с окружающей средой / С. Н. Волкова, Е. Е. Сивак, М. И. Пашкова, В. В. Герасимова // Естественные и математические науки: вопросы и тенденции развития: сб. науч. тр. по итогам междунар. науч.-практ. конф. Инновационный центр развития образования и науки. Красноярск: Инновационный центр развития образования и науки. 2014. С. 9–12.

24. Яковлев А. В., Васюкова Е. О., Пеливан М. А. Моделирование взаимодействия клиента с сервером в информационной системе // Евразийский союз ученых. 2015. № 10–2 (19). С. 184–187.

25. Израилов К. Е. Модель прогнозирования угроз телекоммуникационной системы на базе искусственной нейронной сети // Вестн. ИНЖЭКОНа. Сер.: Технические науки. 2012. № 8 (59). С. 150–153.

P. A. Kurta

Bonch-Bruевич Saint-Petersburg State University of Telecommunications

INTERACTION OF THE USER WITH THE INFORMATION SYSTEM. PART 3. EFFICIENCY EVALUATION

The article is the third in a series devoted to the issues of user interaction with the information system. The problem of calculating and evaluating the effectiveness of interactions is considered. For this, it is proposed to classify the group of compared information services (implementation of the interaction interface and information system) to a certain class for which the interaction model has been built. Then, in terms of the model, it is possible to calculate performance indicators – potency, as a guarantee of the user getting the result; operativeness, like the speed of entering and receiving data; resource preservation, as the moderation of the cognitive load on a person during work. The resulting metric values can be directly used to compare interactions. The article identifies 3 classes of common information services: request, control and gaming. For the first of them, as simple enough for research, an analytical model of interaction is built, and 3 implementation schemes are proposed, which differ in the density of filling the interface forms with input/output fields. Formulas for calculating indicators are displayed. An experiment is carried out to calculate the indicators of the effectiveness of interactions in each of the schemes for a different number of interface elements. The values of the indicators, as well as the root-mean-square efficiency for them and their pairs, are given in tabular and graphical form; also, the ranking of the efficiency of schemes by Pareto is considered. The analysis of the results obtained is carried out and conclusions are drawn that confirm the validity of the formulas obtained. Directions for further research are indicated.

Information system, interface, efficiency, indicators, model

УДК 004.42

Е. Е. Кошелев, С. В. Букунов

*Санкт-Петербургский государственный
архитектурно-строительный университет*

Плагин-синтезатор для цифровой рабочей станции

Приведено описание аудиосинтезатора с собственным графическим интерфейсом пользователя. В плагине реализованы два осциллятора, ADSR-ограничивающая, фильтр нижних частот, фильтр верхних частот и эффект низкочастотной модуляции. Используемые осцилляторы генерируют волны четырех типов: синусоидальную, квадратную, треугольную и пилообразную. Реализована возможность смешения волн двух осцилляторов в любой пропорции. Для реализации каждой составляющей плагина были разработаны необходимые алгоритмы. Для удобства работы с плагином был создан графический пользовательский интерфейс в виде фортепианной клавиатуры, клавиши которой срабатывают по нажатию на левую кнопку мыши. Интерфейс содержит ручки и кнопки для изменения параметров обработки сигнала, а также ручку для регулировки громкости выходного сигнала. Разработанный плагин имеет формат VSTi и может использоваться практически в любой современной цифровой звуковой рабочей станции. Плагин реализован на языке C++ с использованием методологии объектно-ориентированного программирования. Для создания плагина использовалась среда разработки Microsoft Visual Studio и библиотека WDL-OL. В качестве цифровой звуковой рабочей станции для загрузки готового плагина использовалась цифровая аудиостанция Reaper.

Цифровая обработка звука, плагин-синтезатор, цифровая рабочая станция, объектно-ориентированное программирование, графический интерфейс пользователя

В последнее время для обработки и генерации звука стали широко применяться компьютеры, постепенно вытесняя используемые в этих целях физические устройства (микшеры, процессоры эффектов и др.) [1], [2]. Именно компьютерная

обработка звука лежит в основе работы современных цифровых звуковых рабочих станций, которые представляют собой специализированное программное обеспечение, предназначенное для записи, хранения, обработки и воспроизведения