УНИВЕРСИТЕТ ИТМО

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2022 Том 22 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
Jule–August 2022 Vol. 22 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК УНИВЕРСИТЕТ ИТМО
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2022-22-4-751-759
УДК 004.514.62

Программная инструментальная система создания адаптивных пользовательских интерфейсов

Лилия Фаритовна Тагирова[1], Андрей Владимирович Субботин[2], Татьяна Михайловна Зубкова[3]✉

[1,2,3] Оренбургский государственный университет, Оренбург, 460000, Российская Федерация

[1] LG-77@mail.ru, https://orcid.org/0000-0002-3388-9462
[2] aws1998@yandex.ru, https://orcid.org/0000-0001-9922-0297
[3] bars87@mail.ru✉, https://orcid.org/0000-0001-6831-1006

Аннотация

Предмет исследования. Для повышения эффективности работы инженера-конструктора требуется использование систем автоматизации проектирования. В настоящее время средства автоматизированного проектирования являются многофункциональными и имеют расширенный пользовательский интерфейс. В зависимости от объема решаемой задачи и уровня подготовки инженеру-конструктору необходимы не все средства систем автоматизированного проектирования. В этом случае средством повышения производительности труда может служить адаптивный интерфейс, который может настраиваться под конкретного пользователя с учетом его опыта и физиологических особенностей (системный опыт, компьютерная грамотность, опыт работы с подобными программами, машинопись, дальтонизм, память, моторика рук). **Метод.** Характеристики, по которым система оценивает пользователя, имеют разные степени неопределенности, неоднозначности, внутренней противоречивости. Данные характеристики трудно формализуются и очень специфичны. Для выполнения оценки целесообразно использование интеллектуальных систем, базирующихся на нечеткой логике и нечетких множествах. Наиболее приемлемый в данном случае — метод Мамдани, в котором используется минимаксная композиция нечетких множеств. Предложенный механизм включает в себя последовательность действий: фаззификация, нечеткий вывод, композиция, дефаззификация. **Основные результаты.** Разработана программная инструментальная система, которая позволяет формировать интерфейсную часть программного обеспечения с учетом возможностей конкретного пользователя. **Практическая значимость.** Внедрение разработанной программной инструментальной системы позволяет выбрать набор элементов индивидуально для каждого инженера-конструктора и сформировать адаптивный прототип интерфейса прикладной программы. В этом случае появляется возможность улучшить взаимодействие человека и компьютера, сделать его более комфортным, уменьшить время на поиск необходимых функций и количество ошибочный действий, повысить качество выполненной работы.

Ключевые слова

адаптивный интерфейс, искусственный интеллект, экспертная система, нечеткая логика и нечеткие множества, программная инструментальная система

Ссылка для цитирования: Тагирова Л.Ф., Субботин А.В., Зубкова Т.М. Программная инструментальная система создания адаптивных пользовательских интерфейсов // Научно-технический вестник информационных технологий, механики и оптики. 2022. Т. 22, № 4. С. 751–759. doi: 10.17586/2226-1494-2022-22-4-751-759

Software development system for creation adaptive user interfaces

Liliya F. Tagirova[1], Andrey V. Subbotin[2], Tatyana M. Zubkova[3]✉

[1,2,3] Orenburg State University, Orenburg, 460000, Russian Federation

[1] LG-77@mail.ru, https://orcid.org/0000-0002-3388-9462
[2] aws1998@yandex.ru, https://orcid.org/0000-0001-9922-0297
[3] bars87@mail.ru✉, https://orcid.org/0000-0001-6831-1006

Abstract

To improve the efficiency of the design engineer, the use of design automation systems is required. Currently, computer-aided design tools are multifunctional and have an expanded user interface. Depending on the scope of the task to be solved and the level of training, the design engineer does not need all the means of computer-aided design systems. In this case, an adaptive interface can serve as a means of increasing labor productivity, which can be customized for a particular user, taking into account his experience and physiological features (system experience, computer literacy, experience working with such programs, typing, color blindness, memory, hand motility). The characteristics by which a user system is evaluated have different degrees of uncertainty, ambiguity, and internal inconsistency. These characteristics are difficult to formalize and they are very specific. To perform the evaluation, it is advisable to use intelligent systems based on fuzzy logic and fuzzy sets. The most acceptable in this case is the Mamdani method which uses a minimax composition of fuzzy sets. The proposed mechanism includes a sequence of actions: fuzzification, fuzzy inference, composition, defazzification. A software development system has been developed that allows you to form an interface part of the software taking into account the capabilities of a particular user. The implementation of the developed software system allows you to select a set of elements individually for each design engineer and form an adaptive prototype of the application program interface. In this case, it becomes possible to improve the interaction between a person and a computer, make it more comfortable, reduce the time to search for the necessary functions and the number of erroneous actions, and improve the quality of the work done.

Keywords

adaptive interface, artificial intelligence, expert system, fuzzy logic and fuzzy sets, software development system

For citation: Tagirova L.F., Subbotin A.V., Zubkova T.M. Software development system for creation adaptive user interfaces. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2022, vol. 22, no. 4, pp. 751–759 (in Russian). doi: 10.17586/2226-1494-2022-22-4-751-759

Введение

Пользовательский интерфейс — средство, обеспечивающее взаимопонимание человека и компьютера. Потому очень важно сделать такую связь дружественной и интуитивно-понятной.

Проблемам разработки пользовательских интерфейсов посвящено значительное количество отечественных и зарубежных научных трудов, которые используются в различных предметных областях [1–5]. При их проектировании современными авторами предлагаются различные подходы. Часто при разработке интерфейсной части программного обеспечения применяются методы искусственного интеллекта: построение адаптивных интерфейсов на основе генетических алгоритмов [6], экспертные системы, теории нечетких множеств [7–9] и др. Вместе с тем, с целью облегчения работы пользователя с программным средством, задействованы методы системного анализа, синтеза, абстрагирования и построения онтологической модели [10]. Часть ученых при решении проблемы построения адаптивных пользовательских интерфейсов предлагают использование принципа разделения декларативного описания математических моделей и их процедурную интерпретацию [11]. Другими авторами применяются известные модели качества программного обеспечения [12], также используются метафорические или идиоматические подходы при создании или усовершенствовании интерфейсной части программного средства [13].

Также осуществляется проектирование прототипов интерфейса с недетерминированным конечным автоматом [14], и разрабатываются специализированные системы [15, 16]. Имеется опыт непосредственного привлечения потенциальных пользователей к созданию интерфейсной части программного средства [17].

В системах автоматизации проектирования (САПР) пользовательский интерфейс — важная часть системы. Интерфейс входит в состав лингвистического обеспечения и представлен диалоговыми языками. Однако функциональность систем не стоит на месте и постоянно расширяет свои возможности, это в свою очередь отражается на интерфейсе пользователя. Он становится более сложным, непонятным для новичков и не эргономичным, а это сказывается на производительности труда инженера-конструктора. Проблема создания адаптивных интерфейсов для прикладных программ в САПР также актуальна [2, 18].

В отличие от существующих аналогов, предлагаемая программная инструментальная система (ПИС) позволяет подбирать не шаблон целиком, а каждый компонент пользовательского интерфейса (размер шрифта, кнопок, расстояние между кнопками, цветовая гамма, звуковое сопровождение, наличие подсказок и командной строки). Таким образом, шаблон интерфейса универсален для каждого пользователя.

Постановка задачи

В настоящей работе реализовано решение проблемы адаптации интерфейсов к особенностям пользователя на примере инженера-конструктора и его автоматизированного рабочего места. Рабочее место включает в себя прикладные программы для проектирования машиностроительных изделий. Для решения данной задачи выполнена разработка специальной программной системы для создания адаптивных прототипов интерфейсов на основе характеристик пользователя.

Разработанная ПИС имеет возможность создания прототипа интерфейса, адаптированного под характеристики пользователя. ПИС реализована в три этапа: оценка характеристик пользователя; реализация подбора компонентов интерфейса; применение подобранного набора компонентов к интерфейсу прикладной программы.

Для наглядности представления движения информационных потоков при проектировании ПИС построена потоковая модель Data Flow Diagram (DFD) с помощью Case-средства автоматизированного проектирования

Рис. 1. Движение информационных потоков при работе программной инструментальной системы

Fig. 1. The movement of information flows during the operation of the software development system

BP-win. На рис. 1 представлена схема DFD с точки зрения программной системы.

Интерфейсная часть программного обеспечения сформирована на основе оценки характеристик пользователя. Для каждого пользователя подобраны компоненты интерфейса: размер шрифта, размер кнопок, расстояние между кнопками, цветовая гамма, наличие командной строки, наличие звукового сопровождения и наличие подсказок. После оценки характеристик пользователя результаты сохраняются в базе данных программной системы.

Для проведения тестирования пользователей в ПИС разработчиком вносятся контрольно-измерительные материалы, направленные на диагностику сформированности каждой его характеристики.

При подборе компонентов интерфейса под определенного пользователя использован метод экспертного оценивания. Для этого на начальном этапе работы эксперт формирует базу правил на основе продукционной модели знаний.

В ходе работы ПИС выполнено сравнение результатов оценки характеристик пользователя с правилами нечеткой экспертной системы, которая является ядром программного средства. В итоге формируется набор компонентов интерфейса и генерируется прототип интерфейса, который соответствует данному пользователю.

Декомпозиция контекстной диаграммы представлена на рис. 2.

Процесс разработки программного средства включает в себя восемь этапов (рис. 2). На начальном этапе формируется и сохраняется типовая информация. Затем формируется база оценочных материалов, по которым будет производиться оценка характеристик пользователя.

Эксперт формирует структуру экспертной системы, создавая лингвистические переменные и термы. Входные переменные — оцениваемые характеристики пользователя, а выходные — компоненты интерфейса, на основе которых будет формироваться прототип прикладной программы.

Следующий этап — оценка характеристик пользователя, при котором требуется выбрать характеристику и оценить ее с помощью разработанных оценочных материалов. Основной этап — подбор компонентов интерфейса, где загружаются результаты оценки пользователя и база правил нечеткой экспертной системы. В ходе работы экспертной системы вычисляется подходящий набор компонентов интерфейса для конкретного пользователя.

На заключительном этапе происходит непосредственное создание прототипа адаптивного интерфейса прикладной программы, который предоставляется для работы в системе САПР.

Математическая модель экспертной системы

Инструментом для определения компонентов интерфейса в ПИС служит нечеткая ЭС. Основные входные данные — сведения о пользователе. Данные характеризуются различной степенью неопределенности, неоднозначности, внутренней противоречивостью, неполнотой, а также представляют количественные и качественные оценки параметров [19].

Так как данные являются трудно формализованными и специфическими, то при выборе метода искусственного интеллекта использована интеллектуальная система, базирующаяся на нечеткой логике и нечетких множествах.

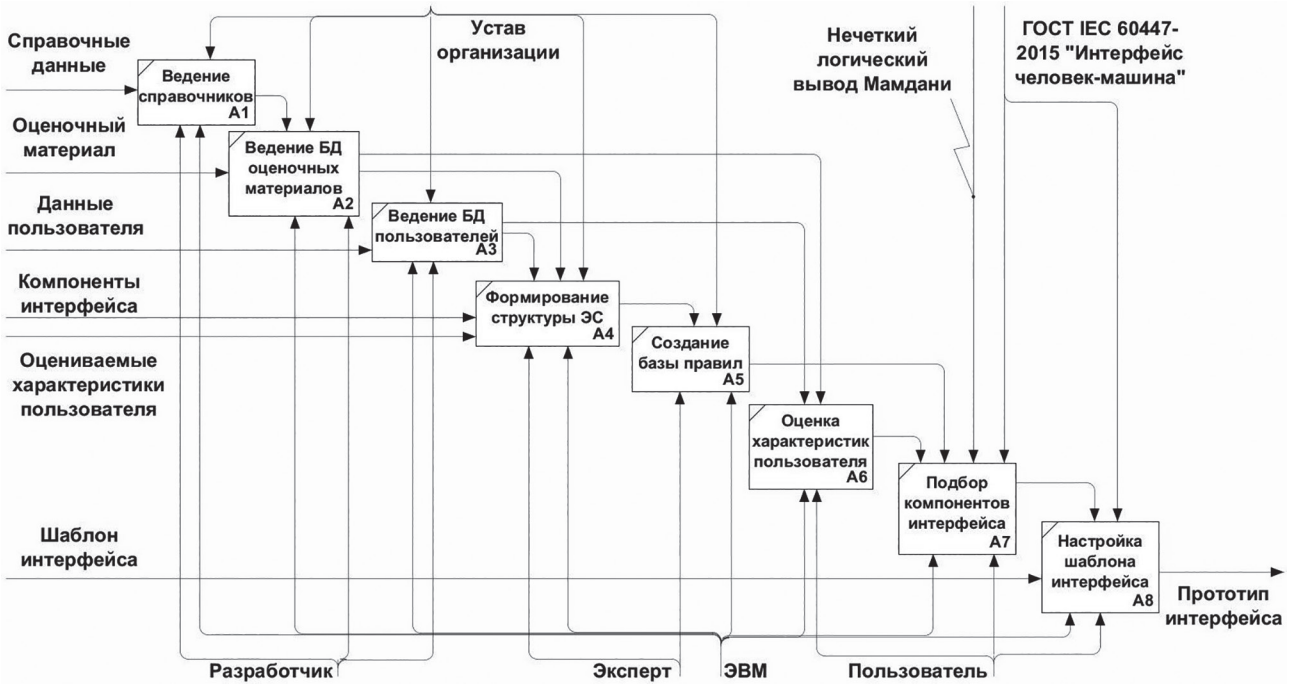

Рис. 2. Декомпозиция контекстной диаграммы.
БД — база данных; ЭВМ — электронно-вычислительная машина; ЭС — экспертная система

Fig. 2. Context diagram decomposition
БД — database; ЭВМ — electronic computer; ЭС — expert system

Из анализа нечеткого понятия «Системный опыт» было сформировано базовое терм-множество, состоящее из трех нечетких переменных: «Низкий», «Средний» и «Высокий», и установлена область рассуждений в виде $X = [0; 100]$ (баллов). Далее была построена функция принадлежности для каждого лингвистического терма из базового терм-множества T.

Существует большое количество стандартных форм кривых для задания функций принадлежности. Самыми распространенными считаются: треугольная, трапецеидальная и гауссова функции принадлежности.

Совокупность функций принадлежности для каждого терма из базового терм-множества T обычно изображается на одном графике. Пример лингвистической переменной «Системный опыт» представлен на рис. 3 в виде трапециевидной функции принадлежности.

Наиболее распространенный способ логического вывода в нечетких системах — механизм Мамдани. Механизм использует минимаксную композицию нечетких множеств и включает в себя последовательность действий [19].

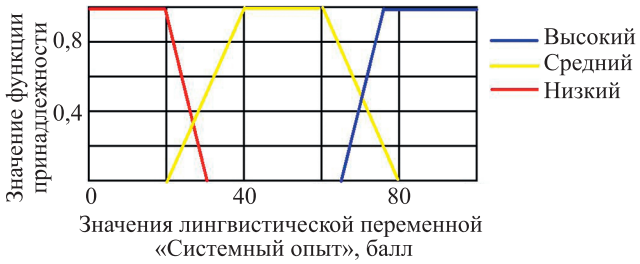

Рис. 3. Графики функций принадлежности значений лингвистической переменной «Системный опыт»

Fig. 3. Graphs of the values belonging functions of the linguistic variable “System experience”

Фаззификация или приведение к нечеткости. Определяются степени истинности, т. е. значения функций принадлежности для левых частей каждого правила (предпосылок или антецедентов). Для базы правил с m правилами обозначим степени истинности как $A_{ik}(x_k)$, $i = 1..m$, $k = 1..n$.

Нечеткий вывод. Определяются уровни «отсечения» для левой части каждого из правил: $\alpha_i = \min_i(A_{ik}(x_k))$, и находятся «усеченные» функции принадлежности: $B_i^*(y) = \min_i(\alpha_i, B_i(y))$.

Композиция или объединение полученных усеченных функций. Используется максимальная композиция нечетких множеств: $\mu(y) = \max_i(B_i^*(y))$, где $\mu(y)$ — функция принадлежности итогового нечеткого множества.

Дефаззификация или приведение к четкости. Существует несколько методов дефаззификации. Например, центроидный метод или метод среднего центра [19].

С учетом введенных понятий построена нечеткая модель, основанная на бинарном нечетком отношении S, которая строится на двух базисных множествах X и Y.

$X = \{x_1\{z_1\}, x_2\{z_2\}, x_3\{z_3\}, \ldots, x_7\{z_7\}\}$ описывает множество компонент интерфейса, где z — множество, характеризующее каждое x.

$Y = \{y_1\{k_1\}, y_2\{k_2\}, y_3\{k_3\}, \ldots, y_7\{k_7\}\}$ — множество характеристик пользователя, где k — множество, характеризующее каждое y.

Элементы универсумов имеют следующий содержательный смысл:

1) x_1 — цветовая гамма (z_1 — черно-белый, z_2 — несколько цветов, z_3 — любые цвета), x_2 — размер шрифта (z_1 — крупный, z_2 — средний, z_3 — мелкий), x_3 — размер кнопок (z_1 — крупные, z_2 — средние, z_3 — мелкие), x_4 — расстояние между кнопками (z_1 — большое, z_2 — среднее, z_3 — малое), x_5 — звуковое сопровождение (z_1 — наличие, z_2 — отсутствие), x_6 — наличие подсказок (z_1 — наличие, z_2 — отсутствие), x_7 — наличие командной строки (z_1 — наличие, z_2 — отсутствие).
2) y_1 — системный опыт (k_1 — высокий, k_2 — средний, k_3 — низкий), y_2 — компьютерная грамотность (k_1 — высокая, k_2 — средняя, k_3 — низкая), y_3 — опыт работы с подобными программами (k_1 — высокий, k_2 — средний, k_3 — низкий), y_4 — машинопись (k_1 — высокая, k_2 — средняя, k_3 — низкая), y_5 — дальтонизм (k_1 — есть, k_2 — нет), y_6 — моторика рук (k_1 — высокая, k_2 — средняя, k_3 — низкая), y_7 — память (k_1 — высокая, k_2 — средняя, k_3 — низкая).

Входными данными являются характеристики пользователей, которые задаются лингвистическими переменными. В табл. 1 представлены все входные лингвистические переменные. Для каждой характеристики определено множество, которое измеряется в баллах. Для образования новых термов использованы процедуры: синтактическая, представляющая собой логическую связку AND (И), и семантическая — $\min(\mu A(x).\ \mu B(x))$.

На примере лингвистической переменной «Системный опыт» описано присвоение значений термов и построение графиков функций принадлежности нечеткого множества (рис. 3). Для остальных входных лингвистических переменных произведена аналогичная процедура.

Далее добавлены лингвистические переменные выходных данных. Они представлены в табл. 2. Для каждой характеристики определено множество из разных единиц измерения.

Для работы механизма нечеткого вывода сформированы продукционные правила. Фрагмент базы правил представлен в табл. 3. Приведено формирования правил для лингвистической переменной «Цветовая гамма», для остальных переменных: «Размер кнопок», «Расстояние между кнопками», «Размер шрифта», «Звуковое сопровождение», «Подсказки», «Командная строка» терм-множества сформированы аналогично.

Экспертная система позволила получить решение на основе описанных правил базы знаний.

Проектирование программной инструментальной системы

Для визуализации работы ПИС использована диаграмма вариантов использования — визуальная модель, отражающая спецификацию программного средства с точки зрения ее функциональности (рис. 4).

Работать с программной системой могут эксперт и пользователь, которые входят в систему с разными правами доступа. Для пользователя подбирается интерфейс.

Таблица 1. Входные лингвистические переменные

Table 1. Input linguistic variables

Название	Терм-множество (T)	Множество-область (X), баллы
Системный опыт	Высокий	65–100
	Средний	35–70
	Низкий	0–45
Компьютерная грамотность	Высокий	70–100
	Средний	35–75
	Низкий	0–40
Опыт работы с подобными программами	Есть	50–100
	Частично	25–60
	Нет	0–30
Машинопись	Быстро	75–150
	Нормально	30–80
	Медленно	0–40
Дальтонизм	Есть	0–1
	Нет	0,9–2
Моторика рук	Высокая	65–100
	Средняя	30–70
	Низкая	0–35
Память	Отличная	75–100
	Умеренная	40–80
	Плохая	0–45

Таблица 2. Выходные лингвистические переменные
Table 2. Output linguistic variables

Название, единица измерения	Терм-множество (*T*)	Множество-область (*X*)
Размер шрифта, пиксел	Крупный	14–18
	Средний	11–15
	Мелкий	8–12
Размер кнопок, коэффициент	Крупный	2–4
	Средний	1,5–2,5
	Мелкий	0–2
Расстояние между кнопками, коэффициент	Большое	2–4
	Среднее	1,5–2,5
	Малое	0–2
Цветовая гамма, коэффициент	Черно-белый	0–1,5
	Несколько цветов	1,4–2,5
	Любые цвета	2,4–4
Звуковое сопровождение, коэффициент	Наличие	0,85–2
	Отсутствие	0–0,9
Наличие подсказок, коэффициент	Наличие	0,85–2
	Отсутствие	0–0,9
Наличие командной строки, коэффициент	Наличие	0,85–2
	Отсутствие	0–0,9

Таблица 3. Нечеткие продукционные правила
Table 3. Fuzzy production rules

Входная лингвистическая переменная			
Если	Условие	то	Терм-множество
Цветовая гамма			
Если	Д = Есть	то	Черно-белый
Если	Д = Нет И СО = Низкий	то	Любые цвета
Если	Д = Нет И СО = Средний	то	Несколько цветов
Если	Д = Нет И СО = Высокий	то	Несколько цветов

Примечание: Д — дальтонизм; СО — системный опыт

Эксперт занимается настройкой экспертной системы. В настройку входят: формирование структуры; для каждой входной переменной подбор оценочного материала; создание базы правил и тестирование созданной экспертной системы с возможностью подробной трассировки расчетов.

Пользователь после авторизации может оценить свои характеристики с помощью оценочного материала, назначенного экспертом, и приступить к формированию личного адаптивного интерфейса прикладной программы, который предоставляется для работы в САПР.

В проектной части ПИС можно выделить три этапа. На первом — производится оценка характеристик пользователя. На втором — независимо от выбора условия формирования прототипа (создать новый или заменить старый прототип), происходит подбор компонентов интерфейса. На третьем — на основе сформированного набора компонентов создается прототип адаптивного интерфейса прикладной программы. Каждый из этих этапов непосредственно взаимодействует с базой данных программной системы.

Практическая реализация программной инструментальной системы

Перед тем как пользователь будет подбирать компоненты интерфейса, эксперт создает структуру экспертной системы (рис. 5). Выбрав пункт меню «Эксперт», задаются лингвистические переменные и термы [20].

Далее эксперт может приступить к формированию базы правил (рис. 6).

После формирования структуры и создания базы правил проводится тестирование экспертной системы (рис. 7). Эксперт, выбрав пункт меню «Подбор компонентов», может установить входные значения. Нажав на кнопку «Рассчитать», ПИС выведет результат подбора компонентов интерфейса в качественных и количественных значениях. Подробные расчеты, выполняе-

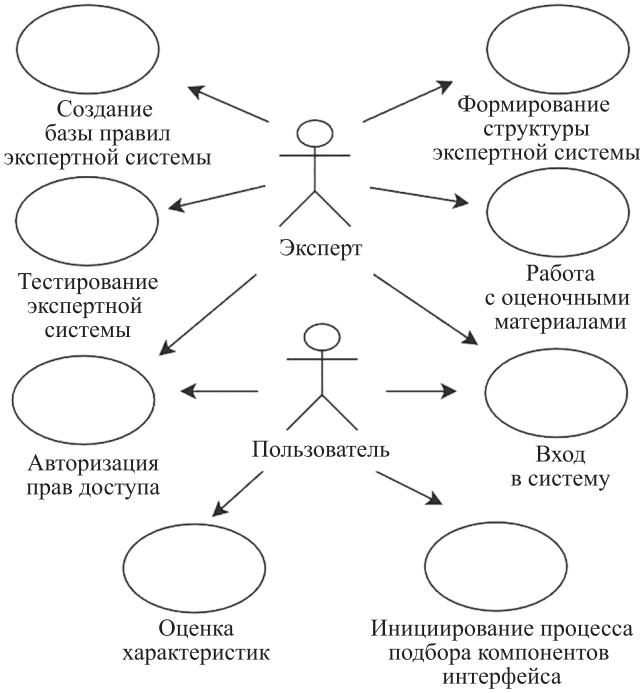

Рис. 4. Диаграмма вариантов использования

Fig. 4. Diagram of use cases

мые экспертной системой, можно просмотреть, нажав на соответствующую кнопку и сохранить в файл (по желанию).

При входе в ПИС пользователю требуется оценить свои характеристики с помощью различных видов диагностики. Например, для оценки качества «Компьютерная грамотность» пользователю предлагается пройти тестирование.

После оценки характеристик пользователь может перейти к подбору компонентов интерфейса и созданию адаптированного прототипа, выбрав пункт меню «Подбор компонентов интерфейса». Для этого требуется выбрать режим создания прототипа и увидеть созданный адаптированный прототип интерфейса прикладной программы (рис. 8).

В приведеном примере (пользователь неопытный) получен прототип, который состоит из следующих компонентов: размер шрифта — средний; размер кнопок — средний; расстояние между кнопками — большое; цветовая гамма — несколько цветов; наличие подсказок — есть; наличие звукового сопровождения — есть, наличие командной строки — нет. Проведенное юзабилити тестирование подтвердило адаптированность интерфейсной части программного обеспечения для определенной аудитории пользователей.

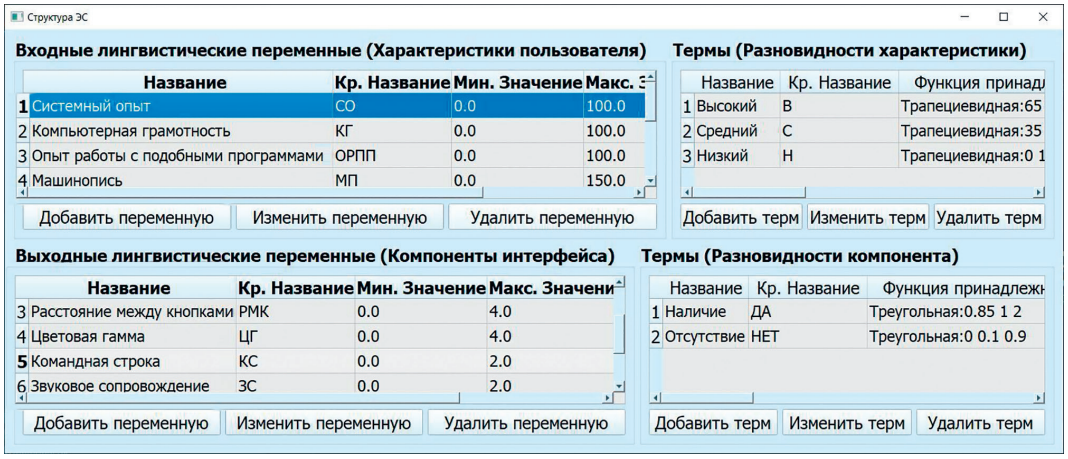

Рис. 5. Структура экспертной системы

Fig. 5. Structure of the expert system

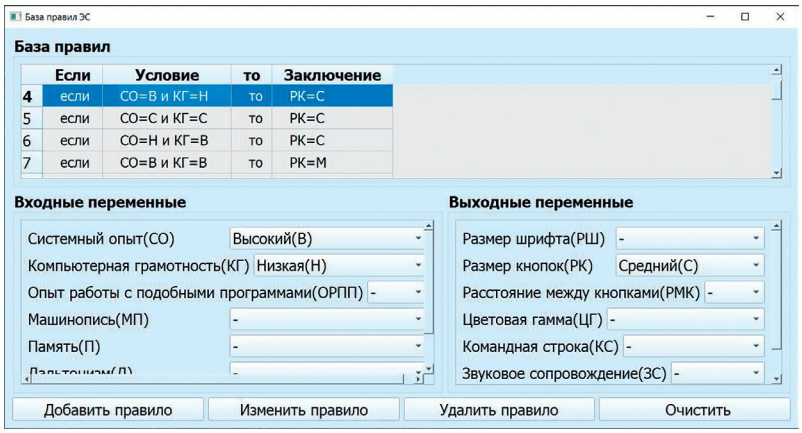

Рис. 6. База правил экспертной системы

Fig. 6. The base of the rules of the expert system

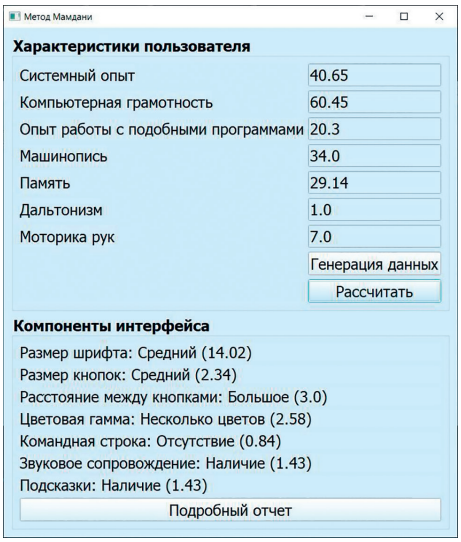

Рис. 7. Тестирование экспертной системы
Fig. 7. Expert system testing

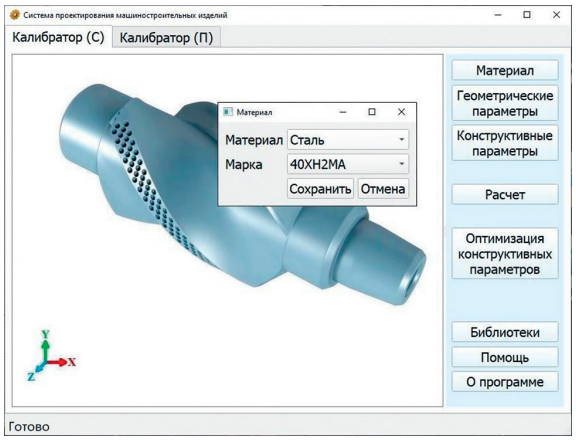

Рис. 8. Фрагмент адаптивного интерфейса
Fig. 8. Fragment of the adaptive interface

Обсуждение результатов

Разработана ПИС, которая учитывает характеристики и возможности пользователя. Для подбора компонентов интерфейса под определенного пользователя использован метод экспертного оценивания. Сформирована база правил на основе продукционной модели знаний. Выполнена оценка результатов полученных характеристик пользователя и правил экспертной системы. В результате создан набор компонентов интерфейса и сгенерирован прототип интерфейса, который соответствует конкретному пользователю.

Так как характеристики пользователя имеют различные степени неопределенности, неоднозначности, внутреннюю противоречивость и др., и являются трудно формализованными и специфическими, то целесообразно использовать интеллектуальные системы, базирующиеся на нечеткой логике и нечетких множествах. Наиболее приемлемым в данном случае является метод Мамдани.

Заключение

Внедрение разработанной экспертной системы позволит просто подобрать набор элементов интерфейса под каждого инженера-конструктора и сформировать адаптивный прототип интерфейса прикладной программы. Данный результат улучшит автоматизированное рабочее место специалиста, а взаимодействие человека и компьютера станет более комфортным и эргономичным.

Литература

1. Будущее разработки ПО — за многообразием пользовательских интерфейсов // Открытые системы. СУБД. 2019. № 2. С. 3–7.
2. Тиханычев О.В. Пользовательские интерфейсы в автоматизированных системах: проблемы разработки // Программные системы и вычислительные методы. 2019. № 2. С. 11–22. https://doi.org/10.7256/2454-0714.2019.2.28443
3. Riaz A., Gregor S., Dewan S., Xu Q. The interplay between emotion, cognition and information recall from websites with relevant and irrelevant images: a neuro-is study // Decision Support Systems. 2018. V. 111. P. 113–123. https://doi.org/10.1016/j.dss.2018.05.004
4. Zubkova T., Tagirova L. Intelligent user interface design of application programs // Journal of Physics: Conference Series. 2019. V. 1278. N 1. P. 012026. https://doi.org/10.1088/1742-6596/1278/1/012026
5. Ben Sassi I., Mellouli S., Ben Yahia S. Context-aware recommender systems in mobile environment: on the road of future research // Information Systems. 2017. V. 72. P. 27–61. https://doi.org/10.1016/j.is.2017.09.001
6. Исмагилова И.М., Валеев С.С. Построение динамических адаптивных интерфейсов информационно-управляющих систем на основе методов искусственного интеллекта // Вестник Уфимского государственного авиационного технического университета. 2018. Т. 2. № 2(80). С. 122–130.
7. Зубкова Т.М., Тагирова Л.Ф., Тагиров В.К. Прототипирование адаптивных пользовательских интерфейсов прикладных программ с использованием методов искусственного интеллекта // Научно-технический вестник информационных технологий,

References

1. The future of software development lies in the variety of user interfaces. *Open Systems.DBMS*, 2019, no. 2, pp. 3–7. (in Russian)
2. Tikhanychev O.V. User interfaces in automated systems: Development issues. *Software systems and computational method*, 2019, no. 2, pp. 11–22. (in Russian). https://doi.org/10.7256/2454-0714.2019.2.28443
3. Riaz A., Gregor S., Dewan S., Xu Q. The interplay between emotion, cognition and information recall from websites with relevant and irrelevant images: a neuro-is study. *Decision Support Systems*, 2018, vol. 111, pp. 113–123. https://doi.org/10.1016/j.dss.2018.05.004
4. Zubkova T., Tagirova L. Intelligent user interface design of application programs. *Journal of Physics: Conference Series*, 2019, vol. 1278, no. 1, pp. 012026. https://doi.org/10.1088/1742-6596/1278/1/012026
5. Ben Sassi I., Mellouli S., Ben Yahia S. Context-aware recommender systems in mobile environment: on the road of future research. *Information Systems*, 2017, vol. 72, pp. 27–61. https://doi.org/10.1016/j.is.2017.09.001
6. Ismagilova I.M., Valeev S.S. Construction of dynamic adaptive interfaces of information-management systems based on methods of artificial intelligence. *Vestnik UGATU*, 2018, vol. 22, no. 2(80), pp. 122–130. (in Russian)
7. Zubkova T.M., Tagirova L.F., Tagirov V.K. Prototyping of adaptive user application programming interfaces by artificial intelligence methods. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2019, vol. 19, no. 4, pp. 680–

механики и оптики. 2019. Т. 19. № 4. С. 680–688. https://doi.org/10.17586/2226-1494-2019-19-4-680-688
8. Зубкова Т.М., Наточая Е.Н. Проектирование интерфейса программного обеспечения с использованием элементов искусственного интеллекта // Программные продукты и системы. 2017. № 1. С. 5–11. https://doi.org/10.15827/0236-235X.030.1.005-011
9. Семенов А.М., Тагирова Л.Ф., Тагиров В.К. Использование нечетких экспертных систем при разработке адаптивных человеко-машинных интерфейсов // Научно-технический вестник Поволжья. 2019. № 7. С. 71–74.
10. Трегубов А.С. Разработка адаптивных контекстозависимых интерфейсов с использованием онтологических моделей // Кибернетика и программирование. 2017. № 6. С. 50–56. https://doi.org/10.25136/2306-4196.2017.6.24747
11. Степанов М.Ф., Степанов А.М. Адаптивный пользовательский интерфейс системы автоматизированного анализа и синтеза алгоритмов управления // Программная инженерия. 2018. Т. 9. № 3. С. 109–122. https://doi.org/10.17587/prin.9.109-122
12. Ахунова Д.Г., Вострых А.В., Курта П.А. Оценка пользовательского интерфейса информационных систем посредствам моделей качества программного обеспечения // Информатизация и связь. 2020. № 2. С. 127–135. https://doi.org/10.34219/2078-8320-2020-11-2-127-135
13. Лукин В.Н., Дзюбенко А.Л., Чечиков Ю.Б. Подходы к разработке пользовательского интерфейса // Программирование. 2020. № 5. С. 16–24. https://doi.org/10.31857/S0132347420050052
14. Vaytsel N.S., Bubareva O.A. Models and methods of computer-aided design of the user interface of software systems // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. 2019. Т. 12. № 1. С. 122–128. https://doi.org/10.14529/mmp190110
15. Саяпин О.В., Тиханычев О.В., Чискидов С.В., Быстракова И.А. Разработка интерфейсов прикладных программ: макетирование или прототипирование // Прикладная информатика. 2020. Т. 15. № 1(85). С. 47–56. https://doi.org/10.24411/1993-8314-2020-10004
16. Вакалюк А.А., Басманов С.Н. Разработка подхода к созданию гибкого пользовательского интерфейса на основе преобразования IDEF0-диаграммы // Современные наукоемкие технологии. 2020. № 5. С. 20–25. https://doi.org/10.17513/snt.38026
17. Беликова С.А. Использование модели деятельности пользователя в предметной области для проектирования пользовательского интерфейса // Информатизация и связь. 2020. № 6. С. 88–91.
18. Шилова Л.А., Пиляй А.И. Естественно языковые интерфейсы для систем автоматизации // Наука и бизнес: пути развития. 2019. № 11(101). С. 94–96.
19. Назаров Д.М. Интеллектуальные системы: основы теории нечетких множеств: учебное пособие для вузов. – 3-е изд., испр. и доп. Москва: Юрайт, 2020. 186 с.
20. Субботин А.В., Тагирова Л.Ф., Тагиров В.К. Инструментальная среда создания прототипов интерфейсов прикладных программ: свидетельство об официальной регистрации программы для ЭВМ RU2021617761. Бюл. 2021. № 5.

688. (in Russian). https://doi.org/10.17586/2226-1494-2019-19-4-680-688
8. Zubkova T.M., Natochaya E.N. Software interface design using elements of artificial intelligence. *Software & Systems*, 2017, no. 1, pp. 5–11. (in Russian). https://doi.org/10.15827/0236-235X.030.1.005-011
9. Semenov A.M., Tagirova L.F., Tagirov V.K. Use of indistinct expert systems when developing adaptive human machine interfaces. *Scientific and Technical Volga region Bulletin*, 2019, no. 7, pp. 71–74. (in Russian)
10. Tregubov A.S. Development of adaptive context-sensitive interfaces using ontological models. *Cybernetics and Programming*, 2017, no. 6, pp. 50–56. (in Russian). https://doi.org/10.25136/2306-4196.2017.6.24747
11. Stepanov M.F., Stepanov A.M. Adaptive user interface for computer-aided control system design. *Software Engineering*, 2018, vol. 9, no. 3, pp. 109–122. (in Russian). https://doi.org/10.17587/prin.9.109-122
12. Akhunova D., Vostrukh A., Kurta P. Evaluation of information systems user interface by means of software quality's models. *Informatization and Communication*, 2020, no. 2, pp. 127–135. (in Russian). https://doi.org/10.34219/2078-8320-2020-11-2-127-135
13. Lukin V.N., Chechikov Y.B., Dzyubenko A.L. Approaches to user interface development. *Programming and Computer Software*, 2020, vol. 46, no. 5, pp. 316–323. https://doi.org/10.1134/S0361768820050059
14. Vaytsel N.S., Bubareva O.A. Models and methods of computer-aided design of the user interface of software systems. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2019, vol. 12, no. 1, pp. 122–128. https://doi.org/10.14529/mmp190110
15. Sayapin O., Tikhanychev O., Chiskidov S., Bystrakova I. Development of application program interfaces: layout or prototype. *Journal of Applied Informatics*, vol. 15, no. 1(85), pp. 47–56. (in Russian). https://doi.org/10.24411/1993-8314-2020-10004
16. Vakalyuk A.A., Basmanov S.N. An approach developing to creating a flexible user interface based on the IDEF0-diagram conversion. *Modern high technologies*, 2020, no. 5, pp. 20–25. (in Russian). https://doi.org/10.17513/snt.38026
17. Belikova S. Using the user activity model in the domain for user interface design. *Informatization and Communication*, 2020, no. 6, pp. 88–91. (in Russian)
18. Shilova L.A., Pilyay A.I. Natural language interfaces for cad systems. *Science and Business: Ways of Development*, 2019, no. 11(101), pp. 94–96. (in Russian)
19. Nazarov D.M. *The Intelligent Systems: Fundamentals of the Fuzzy-Set Theory*. Moscow, Urait Publ., 2020, 186 p. (in Russian)
20. Subbotin A.V., Tagirova L.F., Tagirov V.K. Development environment of the application program interface prototypes. *Certificate of the computer program official registration*. RU2021617761, 2021. (in Russian)

Авторы

Тагирова Лилия Фаритовна — кандидат педагогических наук, доцент, доцент, Оренбургский государственный университет, Оренбург, 460018, Российская Федерация, SC 57210446135, https://orcid.org/0000-0002-3388-9462, LG-77@mail.ru
Субботин Андрей Владимирович — аспирант, Оренбургский государственный университет, Оренбург, 460018, Российская Федерация, https://orcid.org/0000-0001-9922-0297, aws1998@yandex.ru
Зубкова Татьяна Михайловна — доктор технических наук, профессор, профессор, Оренбургский государственный университет, Оренбург, 460018, Российская Федерация, SC 57202282917, https://orcid.org/0000-0001-6831-1006, bars87@mail.ru

Authors

Liliya F. Tagirova — Cand. Sc. (Pedagogy), Associate Professor, Associate Professor, Orenburg State University, Orenburg, 460018, Russian Federation, SC 57210446135, https://orcid.org/0000-0002-3388-9462, LG-77@mail.ru
Andrey V. Subbotin — PhD Student, Orenburg State University, Orenburg, 460018, Russian Federation, https://orcid.org/0000-0001-9922-0297, aws1998@yandex.ru
Tatyana M. Zubkova — D. Sc., Full Professor, Orenburg State University, Orenburg, 460018, Russian Federation, SC 57202282917, https://orcid.org/0000-0001-6831-1006, bars87@mail.ru

Статья поступила в редакцию 18.03.2022
Одобрена после рецензирования 07.06.2022
Принята к печати 20.07.2022

Received 18.03.2022
Approved after reviewing 07.06.2022
Accepted 20.07.2022