УДК 004.5; 004.8
DOI: 10.15827/0236-235X.117.005-011

Дата подачи статьи: 22.07.16
2017. Т. 30. № 1. С. 5–11

ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ ЭЛЕМЕНТОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Т.М. Зубкова, д.т.н., профессор, bars87@mail.ru

(Оренбургский государственный университет, просп. Победы, 13, г. Оренбург, 460018, Россия);

Е.Н. Наточая, к.п.н., доцент, en_ischa@mail.ru

*(Российская академия народного хозяйства и государственной службы при Президенте
Российской Федерации (Оренбургский филиал), ул. Курача, 26, г. Оренбург, 460000, Россия)*

Для разработки качественного ПО необходимо в техническом задании отразить все требования и пожелания заказчика, чтобы у него и у исполнителя сложилось единое представление о будущем программном продукте. Одним из вариантов достижения данного взаимопонимания является разработка прототипа пользовательского интерфейса.

В статье описана методика подбора альтернативного варианта шаблона интерфейса, использующая такие методы искусственного интеллекта, как экспертная оценка и теория нечетких множеств. На основе индивидуальных характеристик пользователей можно разделить на пять групп: новичок, обычный, уверенный, квалифицированный, администратор. Выявлены основные параметры индивидуальных характеристик, по которым можно классифицировать пользователей при проектировании интерфейсов: компьютерная грамотность, системный опыт, опыт работы с подобными программами, машинопись, мышление, память, моторика, дальтонизм, концентрация внимания, эмоциональная устойчивость.

В статье описано программное и математическое обеспечение для решения задач интеллектуального проектирования пользовательского интерфейса. Поставленная задача выполняется в три этапа. На первом этапе – «Формирование и оценка компетентности группы экспертов» – определяются характеристики экспертов. Количественное описание характеристик экспертов основывается на вычислении относительных коэффициентов компетентности по результатам высказываний специалистов о составе экспертной группы. На втором этапе – «Групповая экспертная оценка объектов при непосредственном оценивании» – определяются рекуррентные отношения для итераций. Третий этап – «Построение нечеткой модели на бинарных нечетких отношениях» – оперирует двумя нечеткими множествами: совокупность групп пользователей и множество шаблонов интерфейсов, максимально эффективных для пользователей с данными характеристиками. Входными данными нечеткой модели являются выделенные нечеткие множества, а выходными – степени соответствия шаблонов интерфейса пользователям.

На основе предложенной методики автоматизирован процесс проектирования пользовательского интерфейса с целью повышения объективности и оперативности решений, принимаемых разработчиками ПО.

Ключевые слова: ПО, пользовательский интерфейс, шаблон интерфейса, прототип интерфейса, техническое задание, экспертная оценка, база знаний, нечеткие отношения.

При проектировании программных продуктов необходимо учитывать требования, пожелания, а также знания и возможности потенциального контингента пользователей, которые отражаются в техническом задании на ПО. Для получения единого представления о создаваемом ПО предусматривается активное взаимодействие разработчика с заказчиком. Заказчики разрабатывают концепцию (часто подсознательную и неполную) того, как их приложение будет работать. Разработчикам же необходимо учитывать аппаратные ресурсы, базовое ПО, операционную систему и другое. Кроме того, требуется знать возможности целевой аудитории пользователей, варианты использования ПО или сценарии работы, которые формируются из пользовательских историй. Одним из способов взаимодействия с заказчиком и получения единого видения ПО является разработка прототипа интерфейса [1]. Прототип пользовательского интерфейса представляет собой макет (черновую, пробную версию) интерфейса, разрабатываемый с целью проверки пригодности предлагаемых для применения концепций, технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

Постановка задачи

Задача заключается в разработке программного средства с функцией поддержки принятия решения по подборке альтернативного варианта шаблона интерфейса на основании экспертной оценки и теории нечетких множеств. Для автоматизации принятия решения необходимо разработать шаблоны пользовательского интерфейса с учетом целевой аудитории [2].

Под шаблоном пользовательского интерфейса будем понимать сочетание одного или нескольких элементов, широко используемых для предоставления навигации, команд и содержимого, которые используются большинством приложений и служат основой для разработки собственного пользовательского интерфейса.

Характеристиками контингента пользователей являются их уровень знаний, физические и психологические качества. Каждая из характеристик оценивается по трехбалльной шкале: высокий, низкий и средний уровень развития.

С учетом разных градаций выделенных характеристик пользователи разделены на пять групп (табл. 1).

Таблица 1
Характеристики групп пользователей
Table 1
User group dimensions

Характеристика	Группа пользователей				
	Новичок	Обычный	Уверенный	Квалифицированный	Программист, хакер, администратор
Компьютерная грамотность	0,1	0,4	0,7	0,8	1
Системный опыт	0,1	0,5	0,7	0,8	1
Опыт работы с подобными программами	0,1	0,5	0,6	0,7	0,9
Машинопись	0,7	0,7	0,8	0,9	0,9
Мышление	0,3	0,6	0,7	0,8	0,9
Память	0,2	0,5	0,6	0,8	1
Моторика	0,6	0,7	0,8	0,9	0,9
Дальтонизм	0,1	0,1	0,1	0,1	0,1
Концентрация внимания	0,3	0,7	0,8	0,9	0,1
Эмоциональная устойчивость	0,2	0,5	0,6	0,8	0,9

Декомпозиция функций программной системы представлена в нотации IDEF0 (Integrated DEFinition) на рисунке 1.

Входными данными программной системы являются характеристики групп пользователей, требования заказчика, лицензия разработчика, лицензия заказчика, шаблоны интерфейсов.

В качестве управляющего воздействия выступают Федеральный закон «О персональных данных», ГОСТ19.201-78, экспертные оценки специалистов в данной предметной области, нечеткие отношения.

На выходе программной системы формируется фрагмент технического задания на разработку ПО с прототипом интерфейса.

Реализация поставленной задачи

Входные данные для задачи проектирования интерфейса программных средств характеризуются той или иной степенью неопределенности, обусловленной неполнотой, внутренней противоречивостью, неоднозначностью, и представляют собой приближенные количественные или качественные оценки параметров процессов проектирования и управления проектированием [3].

Таким образом, так как исходные данные задачи трудно формализуемы, целесообразно применить один из методов искусственного интеллекта, основанный на нечеткой логике.

Нечеткие алгоритмы, оперирующие лингвистическими переменными, значения которых задаются нечеткими множествами, удобны для описания слабо формализуемых процессов. Такие алгоритмы

интуитивно более понятны. Их автоматизация позволяет повысить объективность и оперативность решений, принимаемых разработчиком ПО [4].

Методология нечеткого логического вывода достаточно успешно применяется при построении систем управления объектами [5], в частности, при разработке компонентов ПО [6, 7] и минимизации рисков программных проектов [8, 9].

Этап 1. Формирование и оценка компетентности группы экспертов.

Экспертная оценка – процедура получения оценки проблемы на основе мнения специалистов с целью принятия решения. Экспертное оценивание предполагает создание некоего разума, обладающего большими способностями по сравнению с отдельным человеком. Основной сложностью этого метода является подбор экспертов, которые должны иметь опыт в соответствующих решаемым задачам областях. При подборе экспертов следует учитывать личную заинтересованность, которая может стать существенным препятствием для получения объективного суждения.

При формировании группы экспертов на стадии выявления знаний учитывались следующие характеристики:

- компетентность (степень квалификации эксперта в данной области знаний);
- креативность (способность решать творческие задачи);
- отношение к экспертизе (негативное или пассивное отношение, занятость, существенно влияющие на качество работы эксперта в группе);
- конформизм (подверженность влиянию авторитетов, при котором их мнение может подавлять лиц, обладающих более высокой компетентностью);
- коллективизм и самокритичность.

Авторами реализован один из возможных путей количественного описания характеристик эксперта, основанный на вычислении относительных коэффициентов компетентности по результатам высказываний специалистов о составе экспертной группы.

Суть методики сводится к тому, что ряду специалистов предлагается высказать мнение о списочном составе экспертной группы. Если в этом списке появляются лица, не вошедшие в исходный список, им тоже предлагается назвать специалистов для участия в экспертизе. После нескольких этапов будет получен достаточно полный список кандидатов в группу [10].

По результатам опроса составляется матрица, по строкам и столбцам которой записываются фамилии экспертов, а элементами таблицы являются переменные:

$$x_{ij} = \begin{cases} 1, & \text{если } j\text{-й эксперт назвал } i\text{-го;} \\ 0, & \text{если } j\text{-й эксперт не назвал } i\text{-го.} \end{cases}$$

При этом эксперт может включать или не включать себя в экспертную группу (то есть $x_{ij} = 0$ или $x_{ij} = 1$).

По данной таблице можно вычислить относительные коэффициенты компетентности, используя алгоритм решения задач о лидере.

Относительные коэффициенты компетентности h -порядка для каждого эксперта имеют следующий вид:

$$K_i^h = \frac{\sum_{j=1}^m x_{ij} k_j^{h-1}}{\sum_{i=1}^m \sum_{j=1}^m x_{ij} k_j^{h-1}}, i = \overline{1, m}, h = 1, 2, \dots,$$

где m – число экспертов в списке (размерность матрицы $\|x_{ij}\|$); h – номер порядка коэффициента компетентности.

Коэффициенты компетентности нормированы так, что их сумма равна единице:

$$\sum_{i=1}^m k_i^h = 1, h = 1, 2, \dots \quad (1)$$

По формуле (1) можно вычислить значение компетентности для различных порядков, начиная с первого.

При $h = 1$ выражение (1) будет иметь вид

$$K_i^1 = \frac{\sum_{j=1}^m x_{ij}}{\sum_{i=1}^m \sum_{j=1}^m x_{ij}}, i = \overline{1, m}. \quad (2)$$

Таким образом, коэффициент компетентности первого порядка – это относительное число экспертов, высказавшихся за включение i -го эксперта в группу.

Относительный коэффициент компетентности второго порядка получаем из (1) для $h = 2$ при условии, что $k_j^1, j = \overline{1, m}$, определены по (2):

$$K_i^2 = \frac{\sum_{j=1}^m x_{ij} k_j^1}{\sum_{i=1}^m \sum_{j=1}^m x_{ij} k_j^1}, i = \overline{1, m}.$$

Коэффициенты второго порядка представляют собой относительное количество голосов взвешенных коэффициентов компетентности первого порядка.

Последовательно вычисляя относительные коэффициенты компетентности более высокого порядка, можно убедиться, что процесс быстро сходится после 3-4 вычислений, то есть относительные коэффициенты быстро стабилизируются [10].

В общем случае коэффициенты относительной компетентности определяются как

$$k_i = \lim_{h \rightarrow \infty} k_i^h, \sum_{i=1}^n k_i = 1.$$

Этап 2. Групповая экспертная оценка объектов при непосредственном оценивании.

Пусть m экспертов провели оценку n объектов по l показателям. Результаты оценивания представлены величинами x_{ij}^h , где i – номер объекта; j – номер эксперта; h – номер показателя. Величины x_{ij}^h , полученные методом непосредственного оценивания, представляют собой числа из некоторого отрезка числовой оси или баллы.

В качестве групповой оценки для каждого из объектов можно принять среднее взвешенное значение его оценки:

$$x_i = \sum_{h=1}^l \sum_{j=1}^m q_h x_{ij}^h k_j, i = 1, 2, \dots, n,$$

где q_h – коэффициенты весов показателей сравнения объектов; k_j – коэффициенты компетентности экспертов.

Величины q_h и k_j являются нормированными, то есть $\sum_{h=1}^l q_h = 1, \sum_{j=1}^m k_j = 1$.

Коэффициенты q_h могут быть определены экспертным путем как средний коэффициент веса h -го показателя по всем экспертам, то есть $q_h = \sum_{j=1}^m q_{hj} k_j$.

При $h=1$ алгоритм вычисления групповых оценок и коэффициентов компетентности экспертов имеет вид:

а) начальные условия при $t=0$:

$$k_j^0 = \frac{1}{m}, j = 1, m, \text{ то есть начальное значение ко-}$$

эффициентов компетентности для всех экспертов принимается одинаковым;

б) рекуррентные соотношения для $t=1, 2, 3, \dots$ представлены в таблице 2;

Таблица 2
Рекуррентные отношения для итераций
Table 2
Recurrence formula for iterations

Аналитическое описание	Расчетная величина
$x_i^t = \sum_{j=1}^m x_{ij}^t k_j^{t-1},$ $i = 1, n$	Групповая оценка для i -го объекта на t -м шаге на основе индивидуальных оценок x_{ij}
$\lambda^t = \sum_{i=1}^n \sum_{j=1}^m x_{ij}^t x_{ij}$	Нормировочный коэффициент
$k_j^t = \frac{1}{\lambda^t} \sum_{i=1}^n x_{ij}^t x_{ij},$ $j = 1, m-1$	Коэффициенты компетентности j -го эксперта на t -м шаге
$k_m^t = 1 - \sum_{j=1}^{m-1} k_j^t$	Коэффициенты компетентности m -го эксперта из условия нормировки [10]

в) признак окончания итерационного процесса:
 $\max \left(\left| x_i^t - x_i^{t-1} \right| \right) < E.$

Этап 3. Построение нечеткой модели на бинарных нечетких отношениях.

Пусть имеются два множества: совокупность групп пользователей и совокупность шаблонов интерфейсов, которые нужно максимально эффективно подобрать для пользователей с заданными характеристиками. Таким образом, входными данными являются указанные множества, а выходными – степени соответствия шаблонов интерфейсов пользователям.

В общем случае нечетким отношением или, точнее, нечетким k -арным отношением, заданным на множествах (универсумах) $X_1, X_2, \dots, X_k$, называется некоторое фиксированное нечеткое подмножество декартова произведения этих универсумов. Другими словами, если обозначить произвольное нечеткое отношение через Q , то по определению $Q = \{ \langle x_1, x_2, \dots, x_k \rangle, \mu_Q(\langle x_1, x_2, \dots, x_k \rangle) \}$, где $\mu_Q(\langle x_1, x_2, \dots, x_k \rangle)$ – функция принадлежности данного нечеткого отношения, которая определяется как отображение $\mu_Q: X_1 \times X_2 \times \dots \times X_k \rightarrow [0, 1]$. Здесь через $\langle x_1, x_2, \dots, x_k \rangle$ обозначен кортеж из k элементов, каждый из которых выбирается из своего универсума: $x_1 \in X_1, x_2 \in X_2, \dots, x_k \in X_k$.

Бинарное нечеткое отношение задается на базисных множествах X_1, X_2 и определяется как нечеткое отношение $Q = \{ \langle x_i, x_j \rangle, \mu_Q(\langle x_i, x_j \rangle) \}$. Здесь $\mu_Q(\langle x_i, x_j \rangle)$ – функция принадлежности бинарного нечеткого отношения, которая определяется как отображение $\mu_Q: X_1 \times X_2 \rightarrow [0, 1]$, а через $\langle x_i, x_j \rangle$ обозначен кортеж из двух элементов, при этом $x_1 \in X_1, x_2 \in X_2$.

Кроме того, для построения решения задачи необходимы композиции нечетких бинарных отношений.

Пусть Q и R – конечные или бесконечные бинарные нечеткие отношения. Причем нечеткое отношение $Q = \{ \langle x_i, x_j \rangle, \mu_Q(\langle x_i, x_j \rangle) \}$ задано на декартовом произведении универсумов $X_1 \times X_2$, а нечеткое отношение $R = \{ \langle x_j, x_k \rangle, \mu_R(\langle x_j, x_k \rangle) \}$ – на декартовом произведении универсумов $X_2 \times X_3$. Нечеткое бинарное отношение, заданное на декартовом произведении $X_1 \times X_3$ и обозначаемое через $Q \circ R$, называется композицией бинарных нечетких отношений Q и R , а его функция принадлежности определяется следующим выражением:

$$\mu_{Q \circ R}(\langle x_i, x_k \rangle) = \max \left\{ \min \left\{ \mu_Q(\langle x_i, x_j \rangle), \mu_R(\langle x_j, x_k \rangle) \right\} \right\},$$

$$(\forall \langle x_i, x_k \rangle \in X_1 \times X_3).$$

Определенную таким образом композицию бинарных нечетких отношений называют (max-min)-композицией или максиминной сверткой нечетких отношений.

Нечеткое бинарное отношение, заданное на декартовом произведении $X_1 \times X_3$ и обозначаемое че-

рез $Q \circ R$, называется (max-*)-композицией бинарных нечетких отношений Q и R , если его функция принадлежности определяется следующим выражением:

$$\mu_{Q \circ R}(\langle x_i, x_k \rangle) = \max_{x_j \in X_3} \left\{ \mu_Q(\langle x_i, x_j \rangle) * \mu_R(\langle x_j, x_k \rangle) \right\},$$

$$(\forall \langle x_i, x_k \rangle \in X_1 \times X_3).$$

В частности, если в этом выражении вместо операции «*» использовать операцию алгебраического умножения, получим определение (max-prod)-композиции.

Нечеткое бинарное отношение, заданное на декартовом произведении $X_1 \times X_3$ и обозначаемое через $Q \circ R$, называется (min-max)-композицией бинарных нечетких отношений Q и R , если его функция принадлежности определяется следующим выражением:

$$\mu_{Q \circ R}(\langle x_i, x_k \rangle) = \min_{x_j \in X_3} \left\{ \max \left\{ \mu_Q(\langle x_i, x_j \rangle), \mu_R(\langle x_j, x_k \rangle) \right\} \right\},$$

$$(\forall \langle x_i, x_k \rangle \in X_1 \times X_3).$$

С учетом введенных понятий построим нечеткую модель, основанную на двух бинарных нечетких отношениях S и T . Первое из этих нечетких отношений строится на двух базисных множествах X и Y , а второе – на двух базисных множествах Y и Z . Здесь $X = \{x_1, x_2, x_3, x_4, x_5\}$ описывает множество интерфейсов, $Y = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}\}$ – множество характеристик пользователей, а $Z = \{z_1, z_2, z_3, z_4, z_5\}$ – множество групп пользователей.

В данном контексте нечеткое отношение S содержательно описывает характеристики интерфейсов, а T – характеристики пользователей.

Элементы универсумов имеют следующий содержательный смысл:

1) x_1 – шаблон интерфейса 1, x_2 – шаблон интерфейса 2, x_3 – шаблон интерфейса 3, x_4 – шаблон интерфейса 4, x_5 – шаблон интерфейса 5;

2) y_1 – компьютерная грамотность, y_2 – системный опыт, y_3 – опыт работы с подобными программами, y_4 – машинопись, y_5 – мышление, y_6 – память, y_7 – моторика, y_8 – дальтонизм, y_9 – концентрация внимания, y_{10} – эмоциональная устойчивость;

3) z_1 – новичок, z_2 – обычный пользователь, z_3 – уверенный пользователь, z_4 – квалифицированный, z_5 – программист, хакер, администратор.

Для определения соответствия интерфейса группе пользователей воспользовались композициями исходных нечетких отношений. Так, (max-min)- и (max-prod)-композиции дают информацию о степени соответствия шаблона интерфейса группе пользователей, а (min-max)-композиция позволяет определить шаблон, который не подходит для данной группы пользователей [11].

Таким образом, применяются три модели: max-min, max-prod, min-max.

Способы для определения результата композиции нечетких отношений могут быть следующими.

Max-min-композиция, или максимальная нечеткая свертка: $\mu_B(y) = \max_{x \in X} \left\{ \min \left\{ \mu_A(x), \mu_Q(\langle x, y \rangle) \right\} \right\}.$

Max-prod-композиция:

$$\mu_B(y) = \max_{x \in X} \left\{ \left\{ \mu_A(x), \mu_Q(\langle x, y \rangle) \right\} \right\}.$$

Min-max-композиция:

$$\mu_B(y) = \min_{x \in X} \left\{ \max \left\{ \mu_A(x), \mu_Q(\langle x, y \rangle) \right\} \right\}.$$

Max-max-композиция:

$$\mu_B(y) = \max_{x \in X} \left\{ \max \left\{ \mu_A(x), \mu_Q(\langle x, y \rangle) \right\} \right\}.$$

Min-min-композиция:

$$\mu_B(y) = \min_{x \in X} \left\{ \min \left\{ \mu_A(x), \mu_Q(\langle x, y \rangle) \right\} \right\}.$$

Max-average-композиция:

$$\mu_B(y) = 0,5 \cdot \max_{x \in X} \left\{ \mu_A(x) + \mu_Q(\langle x, y \rangle) \right\}.$$

Sum-prod-композиция:

$$\mu_B(y) = f \left(\sum_{x \in X} (\mu_A(x) \cdot \mu_Q(\langle x, y \rangle)) \right).$$

Результаты применения разработанной методики

Рассмотренная методика реализована в программном средстве «Автоматизированная информационная система составления технического задания с экспертной оценкой принятия решения», предназначенном для поддержки руководителя программного проекта при разработке технического задания на ПО с подбором альтернативного варианта интерфейса [12]. Главное окно программного средства показано на рисунке 2.

Рис. 2. Главное окно программного средства

Fig. 2. The software tool main screen

Одной из функций разработанного программного средства является тестирование пользователя, реализованное в форме «Оценка пользователем», в которой можно выбрать характеристику и ее градацию из выпадающих меню (http://www.swsys.ru/uploaded/image/2017_1/2017-1-dop/5.jpg).

Оценка пользователя экспертами (выбор его характеристик и их градаций) основывается на результатах тестирования пользователя и проводится

в форме «Оценка пользователя экспертами» (http://www.swsys.ru/uploaded/image/2017_1/2017-1-dop/6.jpg).

Результаты вычислений выводятся в окне «Анализ» (http://www.swsys.ru/uploaded/image/2017_1/2017-1-dop/7.jpg). По нажатии кнопки «Сохранить отчет» генерируется документ, в котором заполнены следующие разделы: наименование ПО, материально ответственное лицо (руководитель проекта), наименование заказчика, характеристики пользователя, рекомендуемый шаблон интерфейса, назначение и область применения ПО, требования к функциональным характеристикам ПО, требования к составу и параметрам технических средств, требования к информационной и программной совместимости, состав программной документации, стадии, этапы разработки и сроки сдачи ПО.

Во вкладке «Справочники» обеспечивается работа с информацией: должности, тип юридического лица, тип предметной области, тип характеристик пользователя, тип шаблона, физическое лицо, группа пользователей, единицы измерения. Эту информацию можно корректировать, добавлять и удалять.

Заключение

Разработанное программное средство ведет базу знаний о различных категориях пользователей с учетом их компьютерной грамотности, моторики, памяти, мышления, дальтонизма, концентрации внимания и др. На основе этих характеристик разрабатываются различные варианты пользовательского интерфейса, которые согласуются с заказчиком и реализуются в конечном ПО.

Основными результатами работы является программное и математическое обеспечение для решения задач интеллектуального проектирования пользовательского интерфейса ПО с учетом индивидуальных характеристик пользователей.

Решены следующие научно-практические задачи:

- выявлены основные параметры, по которым можно классифицировать пользователей при разработке интерфейсов ПО;

- разработана методика проектирования адаптированных пользовательских интерфейсов с элементами искусственного интеллекта;

- разработана нечеткая модель на бинарных нечетких отношениях для подбора альтернативного шаблона интерфейса ПО.

Предложенная методика позволит автоматизировать процесс проектирования пользовательского интерфейса, конкретизировать техническое задание на ПО, что в конечном итоге повысит объективность и оперативность решений, принимаемых разработчиками.

Литература

1. Брауде Э. Технология разработки программного обеспечения. СПб: Питер, 2004. 655 с.
2. Кулямин В.В. Технологии программирования. Компонентный подход. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2011. 463 с.
3. Таганов А.И., Гильман Д.В. Методологические основы анализа и аттестации уровней зрелости процессов программных проектов в условиях нечеткости: монография. М.: Горячая линия-Телеком, 2014. 168 с.
4. Ишакова Е.Н., Зубкова Т.М. Автоматизация управления рисками программных проектов на основе нечеткого логического вывода // Науч.-технич. вестн. информ. технологий, механики и оптики. 2015. № 5. С. 877–885.
5. Kudinov Yu.I. Synthesis of a fuzzy-logic control system. Jour. of Comp. and Syst. Sc. Intern. 1999, no. 1, pp. 158–164.
6. Chen J., Rine D.C. Training fuzzy logic controller software components by combining adaptation algorithms. Advances in Engineering Software. 2003, no. 3, pp. 125–137.
7. Kandel A., Zhang Y.-Q., Henne M. On use of fuzzy logic technology in operating systems. Fuzzy Sets And Systems. 1998, no. 3, pp. 241–251.
8. Wang J. A fuzzy project scheduling approach to minimize schedule risk for product development. Fuzzy Sets And Systems. 2002, no. 2, pp. 99–116.
9. Carr V., Tah J.H.M. A fuzzy approach to construction project risk assessment and analysis: construction project risk management system. Advances in Engineering Software. 2001, no. 10–11, pp. 847–857.
10. Джарратано Д., Райли Г. Экспертные системы: принцип разработки и программирование. 4-е изд. М.: Вильямс, 2007. 1152 с.
11. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб: БХВ-Петербург, 2005. 736 с.
12. Зубкова Т.М., Колобов А.Н. Автоматизированная информационная система составления технического задания с экспертной оценкой принятия решения. Свид. о гос. регистр. прог. для ЭВМ № 2016616452. Заявл. 18.04.2016. Опубл. 10.06.2016.

SOFTWARE INTERFACE DESIGN USING ELEMENTS OF ARTIFICIAL INTELLIGENCE

T.M. Zubkova¹, Dr.Sc. (Engineering), Professor, bars87@mail.ru

E.N. Natochaya², Ph.D. (Education), Associate Professor, en_ischa@mail.ru

¹ Orenburg State University, Pobedy Ave. 13, Orenburg, 460018, Russian Federation

² Russian Presidential Academy of National Economy and Public Administration (Orenburg branch), Curacha St. 26, Orenburg, 460000, Russian Federation

Abstract. In order to develop high-quality software it is necessary to reflect all customer requirements in the specification, thus, to have a global view on the future software for customers and performers. One of the options to achieve mutual understanding is to develop a prototype of a user interface.

The article describes the methods of selecting an alternative version of the interface template using such artificial intelligence methods as expert evaluation and the fuzzy-set theory. Users might be divided into five groups on the basis of individual characteristics (a newbie, usual, experienced, skilled, an administrator). The article defines the basic parameters of individual characteristics which may help to classify users when designing interfaces (computer literacy, systematic experience, experience of working with similar programs, typing, thinking, memory, motor skills, blindness, concentration, emotional stability).

The paper describes mathematical support and software for solving the problems of intelligent user interface design. Task implementation is performed in three stages. The first stage is "Forming and assessing expert group competence". It defines the characteristics of experts. A quantitative description of experts' characteristics is based on the calculation of relative ratios of competence according to the results of experts' statements on the Advisory group. The second stage is "Group expert assessment of the object with direct assessment". It determines recurrent relations for iterations. The third phase is "Building a fuzzy model on fuzzy binary relations". It operates by two fuzzy sets: a set of user groups and a variety of interface templates that are maximally effective for users with these characteristics. Fuzzy model input data are selected fuzzy sets, the output data are the degrees of matching interface patterns to users.

The user interface design process is automated on the basis of the proposed methodology in order to improve objectivity and optimize decisions taken by software developers.

Keywords: software, user interface, template interface, interface prototype, technical specification, expert assessment, knowledge base, fuzzy relations.

References

1. Braude E. *Tekhnologiya razrabotki programmnogo obespecheniya* [The Software Development Technology]. St. Petersburg, Piter Publ., 2004, 655 p.
2. Kulyamin V.V. *Tekhnologii programmirovaniya* [Programming Technologies. The component approach]. Moscow, Internet-University of Information Technology, BINOM. Laboratoriya Znaniy Publ., 2011, 463 p.
3. Taganov A.I., Gilman D.V. *Metodologicheskie osnovy analiza i attestatsii urovney zrelosti protsessov programmykh proektov v usloviyakh nechetkosti* [Methodological framework for analysis and assessment of software project process maturity levels in terms of fuzziness]. Monograph, Moscow, Goryachaya liniya-Telecom Publ., 2014, 168 p.
4. Ishakova E.N., Zubkova T.M. Risk management automation of software projects based on fuzzy inference. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Jour. of Information Technologies, Mechanics and Optics]. 2015, no. 5, pp. 877–885 (in Russ.).
5. Kudinov Yu.I. Synthesis of a fuzzy-logic control system. *Jour. of Computer and Systems Sciences Int.* 1999, no. 1, pp. 158–164.
6. Chen J., Rine D.C. Training fuzzy logic controller software components by combining adaptation algorithms. *Advances in Engineering Software*. 2003, no. 3, pp. 125–137.
7. Kandel A., Zhang Y.-Q., Henne M. On use of fuzzy logic technology in operating systems. *Fuzzy Sets And Systems*. 1998, no. 3, pp. 241–251.
8. Wang J. A fuzzy project scheduling approach to minimize schedule risk for product development. *Fuzzy Sets And Systems*. 2002, no. 2, pp. 99–116.
9. Carr V., Tah J.H.M. A fuzzy approach to construction project risk assessment and analysis: construction project risk management system. *Advances in Engineering Software*. 2001, no. 10–11, pp. 847–857.
10. Giarratano J.C., Riley G.D. *Expert systems: the principle of development and programming*. 4th ed. Course Technology Publ., 2004, 288 p. (Russ.ed.: Moscow, Vilyams Publ., 2007, 1156 p.).
11. Leonenkov A.V. *Nechetkoe modelirovanie v srede MATLAB i fuzzyTECH* [Fuzzy modeling in MATLAB and fuzzyTECH]. St. Petersburg, BHV-Peterburg Publ., 2005, 736 p.
12. Zubkova T.M., Kolobov A.N. *Avtomatizirovannaya informatsionnaya sistema sostavleniya tekhnicheskogo zadaniya s ekspertnoy otsenkoy prinyatiya resheniya* [Automated Information System for Creating Technical Task with Expert Assessment of Decision-Making]. Software State Registration Certificate no. 2016616452. 2016.

Примеры библиографического описания статьи

1. Zubkova T.M., Natochaya E.N. Проектирование интерфейса программного обеспечения с использованием элементов искусственного интеллекта // Программные продукты и системы. 2017. Т. 30. № 1. С. 5–11; DOI: 10.15827/0236-235X.117.005-011.

2. Zubkova T.M., Natochaya E.N. Software interface design using elements of artificial intelligence. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 1, pp. 5–11 (in Russ.); DOI: 10.15827/0236-235X.117.005-011.