

## Реализация нечёткого выбора оборудования в системе проектирования информационной сети\*

Е. Н. Чуйкова

*Рассматривается реализация процесса выбора оборудования в системе проектирования информационной сети на основе нечётких критериев. Выбор осуществляется в реляционной базе данных посредством нечётких SQL-запросов, формируемых в программной оболочке системы проектирования. Предложен алгоритм формирования нечётких SQL-запросов. Приведён пример построения нечёткого запроса для выбора наилучшего по соотношению цены и качества оборудования. Представлен фрагмент графического интерфейса системы, содержащий окно выбора оборудования. Описанный способ выполнения нечётких запросов применим для реализации нечёткого поиска в различных предметных областях, не требует от пользователя дополнительных знаний в нечёткой логике, автоматизирует процесс формирования функций принадлежности термов лингвистической переменной. Направление дальнейшей работы будет связано с расширением возможностей интерфейса системы по непосредственному конструированию нечётких запросов пользователем.*

**Ключевые слова:** реляционная база данных, SQL-запросы, нечёткая логика, лингвистическая переменная, нечёткое множество, функция принадлежности.

**Введение.** Одной из задач, требующих решения при проектировании информационных сетей, является выбор сетевого оборудования, используемого для построения сети. При наличии на рынке широкого, постоянно обновляемого ассортимента аналогичного оборудования, предлагаемого разными производителями, решение задачи выбора надёжного, и в то же время не слишком дорогого оборудования оказывается чрезвычайно сложным. Сегодня проектировщики сетей при выборе оборудования полагаются на имеющийся опыт и личные пристрастия. Предлагается оснастить систему проектирования информационных сетей средствами, позволяющими оптимизировать выбор оборудования на основе автоматизированного нечёткого поиска в реляционной базе данных, хранящей справочную информацию для системы. Нечёткий поиск наилучшего по соотношению цены и качества сетевого оборудования базируется на теории нечётких множеств [1—5] и осуществляется посредством нечётких SQL-запросов [6, 7].

Существующие решения по реализации нечёткого поиска в реляционной базе данных предполагают хранение вместе с информацией о характеристиках объектов значений функций принадлежности соответствующим нечётким множествам и выполнение нечёткого поиска с использованием этих значений. Такое решение существенно замедляет и усложняет ввод данных в базу данных и увеличивает объём хранимой информации. Для устранения указанных недостатков нечёткие SQL-запросы реализуются в программной оболочке системы с использованием средств графического интерфейса, позволяющих автоматизировать процедуру формирования нечётких запросов. Подобная идея используется в [8] при создании программного шлюза, выполняющего трансляцию операторов реляционной алгебры в инструкции стандартного языка SQL.

**Постановка задачи.** В реляционной БД системы проектирования хранится фактическая информация о технических характеристиках, ценах, сроках гарантии на сетевое оборудование различных типов. При построении конкретной информационной сети необходимо выполнять выбор из БД наилучшего по соотношению цены и качества оборудования с требуемыми характеристиками. Результат выбора должен представляться в виде экранной таблицы, содержащей перечень соответствующего оборудования с указанием всех его параметров. Выбор по критерию наилучшего соотношения цены и качества предполагает осуществление нечёткого поиска с использованием

\* Работа выполнена в рамках инициативной НИР.

лингвистических переменных (ЛП), соответствующих используемым нечётким понятиям «невысокая цена», «высокое качество» [5, 9–11]. При этом необходимо определить универсальные множества этих переменных, наборы базовых термов и задать функции принадлежности каждого терма с учётом предъявляемых к ним требований [5, 10].

Выборка данных в реляционной БД выполняется посредством SQL-запросов, поэтому необходимо определить порядок преобразования нечёткого поискового запроса в инструкцию стандартного языка SQL. При этом требуется сформировать фрагменты SQL-запроса, реализующие вычисление значений функций принадлежности термов ЛП и проверку условий отбора записей из таблиц БД. Отбираются записи, для которых степень принадлежности  $\mu_{T_k}$  соответствующему терму  $T_k$  из базового терм-множества  $T$  превышает заданное значение  $\alpha$ :  $\mu_{T_k} \geq \alpha$ ,  $0 \leq \alpha \leq 1$  [7].

**Алгоритм формирования нечётких запросов.** Для решения задачи нечёткого поиска необходимо задать лингвистические переменные «Цена сетевого оборудования» и «Качество сетевого оборудования». Универсальное множество ЛП «Цена сетевого оборудования»  $U = [\min, \max]$ , где  $\min$ ,  $\max$  — минимальная и максимальная цена данного вида оборудования соответственно; базовое терм-множество  $T = \{\text{«низкая»}, \text{«ниже средней»}, \text{«средняя»}, \text{«выше средней»}, \text{«высокая»}\}$ .

Качество сетевого оборудования опосредованно можно оценить по сроку гарантии на него или сроку службы: чем больше срок гарантии или срок службы, тем выше качество. Универсальное множество ЛП «Качество сетевого оборудования»  $U = [0, N]$ , базовое терм-множество  $T = \{\text{«низкое»}, \text{«среднее»}, \text{«высокое»}\}$ . Максимальный срок гарантии  $N$  и цена оборудования выбираются для каждого вида оборудования из БД системы.

Пусть термы лингвистических переменных представлены параметрическими нечёткими числами с трапециевидными и треугольными функциями принадлежности как наиболее распространёнными [12]. Тогда функция принадлежности терма  $T_1$  имеет вид:

$$\mu_{T_1}(x) = \begin{cases} 0, & x < a_{11}, \\ 1, & a_{11} \leq x \leq a_{12}, \\ f_{12}(x), & a_{12} < x \leq a_{13}, \\ 0, & x > a_{13}. \end{cases} \quad (1)$$

Функция принадлежности терма  $T_n$  имеет вид:

$$\mu_{T_n}(x) = \begin{cases} 0, & x < a_{n1}, \\ f_{n1}(x), & a_{n1} \leq x < a_{n2}, \\ 1, & a_{n2} \leq x \leq a_{n3}, \\ 0, & x > a_{n3}. \end{cases} \quad (2)$$

Функции принадлежности остальных термов  $T_k$  имеют вид:

$$\mu_{T_k}(x) = \begin{cases} 0, & x < a_{k1}, \\ f_{k1}(x), & a_{k1} \leq x \leq a_{k2}, \\ f_{k2}(x), & a_{k2} < x \leq a_{k3}, \\ 0, & x > a_{k3}. \end{cases} \quad (3)$$

Здесь  $n$  — количество термов ЛП;  $a_{11}, a_{n3}$  — соответственно левая и правая границы универсального множества ЛП;  $a_{i1} < a_{i2} < a_{i3}$  — параметры нечёткого числа, задающего терм  $T_i$ ;

$f_{i1}(x) = \frac{x - a_{i1}}{a_{i2} - a_{i1}}$ ,  $f_{i2}(x) = \frac{a_{i3} - x}{a_{i3} - a_{i2}}$  — линейные функции, определяющие форму терма  $T_i$ . В общем

случае функции  $f_{ij}$  могут задавать произвольные зависимости (квадратичную, экспоненциальную и т. д.), использование которых потребует незначительных изменений в программном коде.

Анализ прайс-листов на сетевое оборудование на интернет-сайте price.ru показал, что цены на однотипные устройства со схожей функциональностью сопоставимы и равномерно распределены в диапазоне изменения цен. В этом случае функции принадлежности базовых термов для ЛП «Цена сетевого оборудования» могут выглядеть так, как представлено на рис. 1.

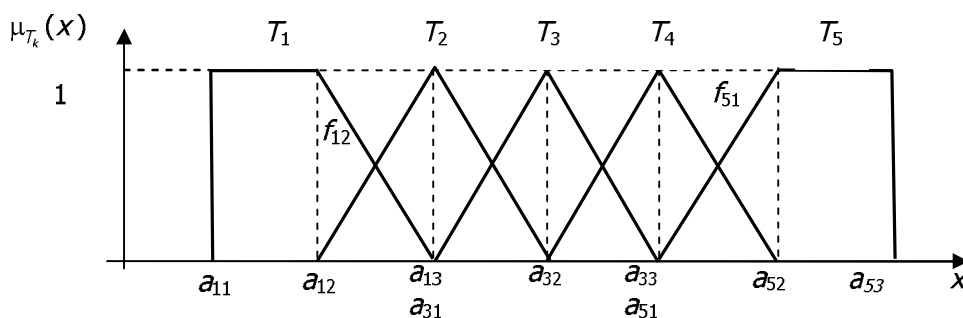


Рис. 1. Функции принадлежности базовых термов ЛП «Цена сетевого оборудования»:  $T_1$  — низкая;  $T_2$  — ниже средней;  $T_3$  — средняя;  $T_4$  — выше средней;  $T_5$  — высокая;  $a_{11}, a_{12}, a_{13}, a_{31}, a_{32}, a_{33}, a_{51}, a_{52}, a_{53}$  — параметры термов  $T_1, T_3, T_5$  соответственно;  $f_{12}, f_{51}$  — функции, определяющие форму термов  $T_1, T_5$  соответственно

При неравномерном распределении значений цены оборудования основания термов ЛП могут определяться таким образом, чтобы каждое из них включало приблизительно равное количество значений цены, имеющих в БД, и дисперсии значений были близки. Но и в этом случае допустимо использовать определение термов ЛП, приведённое на рис. 1, поскольку задание ЛП осуществляется на основе субъективных оценок и не связано с распределением значений рассматриваемого параметра.

Функции принадлежности базовых термов для ЛП «Качество сетевого оборудования» определяются аналогично (рис. 2).

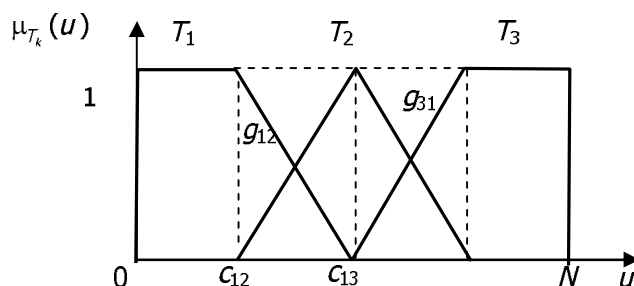


Рис. 2. Функции принадлежности базовых термов ЛП «Качество сетевого оборудования»:  $T_1$  — низкое;  $T_2$  — среднее;  $T_3$  — высокое;  $c_{12}, c_{13}$  — параметры терма  $T_1$ ;  $g_{12}, g_{31}$  — линейные функции, определяющие форму термов  $T_1, T_3$  соответственно

Поиск в БД сетевого оборудования с заданными характеристиками, цена которого, например, низкая или ниже средней и качество (срок гарантии) среднее или высокое, реализуется с помощью нечётких запросов, которые необходимо трансформировать в стандартные инструкции SQL.

Если термы ЛП заданы формулами (1)–(3), то условное выражение в предложении WHERE SQL-запроса, проверяющее степень принадлежности объекта БД терму ЛП, выглядит следующим образом:

для терма  $T_1$

$$([name] \geq a_{11} \text{ AND } [name] \leq a_{12}) \text{ OR } ([name] > a_{12} \text{ AND } [name] < a_{13} \text{ AND } f_{12}([name]) \geq \alpha);$$

для термина  $T_n$

$$([name] > a_{n1} \text{ AND } [name] < a_{n2} \text{ AND } f_{n1}([name]) \geq \alpha) \text{ OR } ([name] \geq a_{n2});$$

для термина  $T_k$

$$([name] \geq a_{k1} \text{ AND } [name] \leq a_{k2}) \text{ AND } f_{k1}([name]) \geq \alpha \text{ OR};$$

$$([name] > a_{k2} \text{ AND } [name] < a_{k3} \text{ AND } f_{k2}([name]) \geq \alpha),$$

где  $name$  — имя поля из соответствующей таблицы БД, по значению которого вычисляется степень принадлежности терму ЛП.

Вычисление значений параметров  $a_{ij}$  и функций  $f_{jk}$  выполняется с помощью подзапросов. Так, фрагмент SQL-запроса, определяющий значение функции  $f_d$  имеет вид:

$$(name - (SELECT <выражение, вычисляющее  $a_d$ > FROM Table)) / (SELECT <выражение, вычисляющее  $a_d - a_d$ > FROM Table),$$

где  $Table$  — имя таблицы БД, из которой выбирается информация.

Сложное условие, проверяющее принадлежность нескольким термам, формируется посредством соединения представленных выше условных выражений с помощью операторов  $AND$  и  $OR$ . Так, условие «цена низкая или ниже средней» задаётся выражением  $q_1 \text{ OR } q_2$ , где  $q_1$  — истинность высказывания «цена низкая»,  $q_2$  — истинность высказывания «цена ниже средней». Высказывание  $q_i$  истинно, если  $\mu_{T_i}(x_0) \geq \alpha$  ( $T_i$  —  $i$ -й терм ЛП «Цена»;  $x_0$  — выбранное из БД значение цены). Аналогично формируется условное выражения SQL-запроса для оценки качества оборудования. Тогда условие «цена низкая или ниже средней и качество (срок гарантии) среднее или высокое» имеет вид:

$$(q_1 \text{ OR } q_2) \text{ AND } (q_3 \text{ OR } q_4),$$

где  $q_3$  — истинность высказывания «качество среднее»,  $q_4$  — истинность высказывания «качество высокое».

SQL-запрос, позволяющий получить параметры требуемого оборудования, выглядит следующим образом:

$$\begin{aligned} &SELECT \text{ имя\_поля\_1, имя\_поля\_2, ..., имя\_поля\_m} \\ &FROM \text{ имя\_таблицы} \text{ WHERE } q \text{ AND } (q_1 \text{ OR } q_2) \text{ AND } (q_3 \text{ OR } q_4), \end{aligned} \quad (4)$$

где  $\text{имя\_поля\_1, имя\_поля\_2, ..., имя\_поля\_m}$  — набор полей, входящих в результирующий набор данных;  $\text{имя\_таблицы}$  — имя таблицы базы данных, из которой выбираются данные;  $q$  — условное выражение, задающее требуемые значения характеристик выбираемого оборудования.

Таким образом, алгоритм построения SQL-запроса, реализующего нечёткий поиск в БД, включает следующие шаги:

1. Сформулировать запрос на выборку данных вида (4).
2. Определить условное выражение  $q$  запроса, указав требуемые характеристики выбираемого объекта.
3. Сформулировать условия  $q_i$  согласно формулам (1) — (3).
4. Построить подзапросы для вычисления параметров  $a_{ij}$  и функций  $f_{jk}$ , используемых в условиях  $q_i$ . В запросе (4) заменить условие  $q$ , параметры и функциональные символы в условиях  $q_i$  соответствующими подзапросами.

**Реализация нечёткого поиска оборудования в системе проектирования информационной сети.** Рассмотрим пример формирования SQL-запроса на поиск в БД системы оборудования, оптимального по соотношению цены и качества.

Каждый вид сетевого оборудования характеризуется своим набором параметров, по которым выполняется поиск требуемой конфигурации. Так, для выбора маршрутизатора или коммута-

тора определяющей характеристикой является количество и типы поддерживаемых ими интерфейсов.

Построим SQL-запрос, позволяющий выбрать из БД маршрутизаторы с заданными характеристиками, цена которых низкая или ниже средней и качество среднее или высокое.

1. Сформулируем предварительный запрос вида (4):

$SELECT * FROM \text{Маршрутизаторы} WHERE q AND (q_1 OR q_2) AND (q_3 OR q_4);$

2. Определим условие  $q$ .

$\text{Число\_портов} = \text{quantityPort} AND \text{Тип\_порта} = \text{typePort},$

где Число\_портов, Тип\_порта — имена полей таблицы «Маршрутизаторы»,  $\text{quantityPort}$  — требуемое количество интерфейсов,  $\text{typePort}$  — требуемый тип интерфейса маршрутизатора.

3. Сформулируем условия  $q_1$  — цена низкая,  $q_2$  — цена ниже средней,  $q_3$  — качество среднее,  $q_4$  — качество высокое:

$q_1 = ([\text{Цена}] \geq a_{11} AND [\text{Цена}] \leq a_{12}) OR ([\text{Цена}] > a_{12} AND [\text{Цена}] < a_{13} AND f_{12}(\text{Цена}) \geq 0.5);$

$q_2 = ([\text{Цена}] \geq a_{21} AND [\text{Цена}] \leq a_{22}) AND f_{21}([\text{Цена}]) \geq 0.5) OR$

$([\text{Цена}] > a_{22} AND [\text{Цена}] < a_{23} AND f_{22}([\text{Цена}]) \geq 0.5);$

$q_3 = ([\text{Срок\_гар}] \geq c_{21} AND [\text{Срок\_гар}] \leq c_{22}) AND g_{21}([\text{Срок\_гар}]) \geq 0.5) OR$

$([\text{Срок\_гар}] > c_{22} AND [\text{Срок\_гар}] < c_{23} AND g_{22}([\text{Срок\_гар}]) \geq 0.5);$

$q_4 = ([\text{Срок\_гар}] > c_{31} AND [\text{Срок\_гар}] < c_{32} AND g_{31}([\text{Срок\_гар}]) \geq 0.5) OR$

$([\text{Срок\_гар}] \geq c_{32}).$

4. Построим подзапросы для вычисления параметров  $a_{ij}$ ,  $c_{kl}$  и функций  $f$ ,  $g$ , используемых в условиях  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ .

$a_{11} = SELECT MIN(\text{Цена}) FROM \text{Маршрутизаторы} WHERE q;$

$a_{12} = a_{21} = SELECT (MAX(\text{Цена}) - MIN(\text{Цена}))/6 + MIN(\text{Цена})$

$FROM \text{Маршрутизаторы} WHERE q;$

$a_{13} = a_{22} = SELECT 2*(MAX(\text{Цена}) - MIN(\text{Цена}))/6 + MIN(\text{Цена})$

$FROM \text{Маршрутизаторы} WHERE q;$

$a_{23} = SELECT 3*(MAX(\text{Цена}) - MIN(\text{Цена}))/6 + MIN(\text{Цена}) FROM \text{Маршрутизаторы} WHERE q;$

$f_{12}(\text{Цена}) = 2*(SELECT (MAX(\text{Цена}) - MIN(\text{Цена}))/6 + MIN(\text{Цена})$

$FROM \text{Маршрутизаторы} WHERE q) - \text{Цена})/$

$(SELECT (MAX(\text{Цена}) - MIN(\text{Цена}))/6 FROM \text{Маршрутизаторы} WHERE q);$

$f_{21}(\text{Цена}) = (\text{Цена} - (SELECT (MAX(\text{Цена}) - MIN(\text{Цена}))/6 + MIN(\text{Цена})$

$FROM \text{Маршрутизаторы} WHERE q))/(SELECT (MAX(\text{Цена}) - MIN(\text{Цена}))/6$

$FROM \text{Маршрутизаторы} WHERE q);$

$f_{22}(\text{Цена}) = ((SELECT 3*(MAX(\text{Цена}) - MIN(\text{Цена}))/6 + MIN(\text{Цена}) FROM$

$\text{Маршрутизаторы} WHERE q) - \text{Цена})/(SELECT (MAX(\text{Цена}) - MIN(\text{Цена}))/6 FROM \text{Маршрутизаторы} WHERE q);$

$c_{21} = N/4; c_{22} = c_{31} = N/2; c_{23} = c_{32} = 3*N/4;$

$g_{21}(\text{Срок\_гар}) = (\text{Срок\_гар} - (SELECT MAX(\text{Срок\_гар})/4 FROM \text{Маршрутизаторы}$

$WHERE q))/(SELECT (MAX(\text{Срок\_гар})/4 FROM \text{Маршрутизаторы}$

$WHERE q);$

$g_{22}(\text{Срок\_гар}) = 3*(SELECT MAX(\text{Срок\_гар})/4 FROM \text{Маршрутизаторы}$

$WHERE q) - \text{Срок\_гар})/(SELECT (MAX(\text{Срок\_гар})/4 FROM \text{Маршрутизаторы} WHERE q).$

После подстановки всех полученных выражений получим окончательный SQL-запрос. Результатом выполнения запроса является таблица, содержащая перечень выбранного оборудования с указанием его характеристик, цены, предприятий-производителей.

Одно из окон системы, в котором пользователь имеет возможность осуществить нечёткий выбор оборудования, представлено на рис. 3.

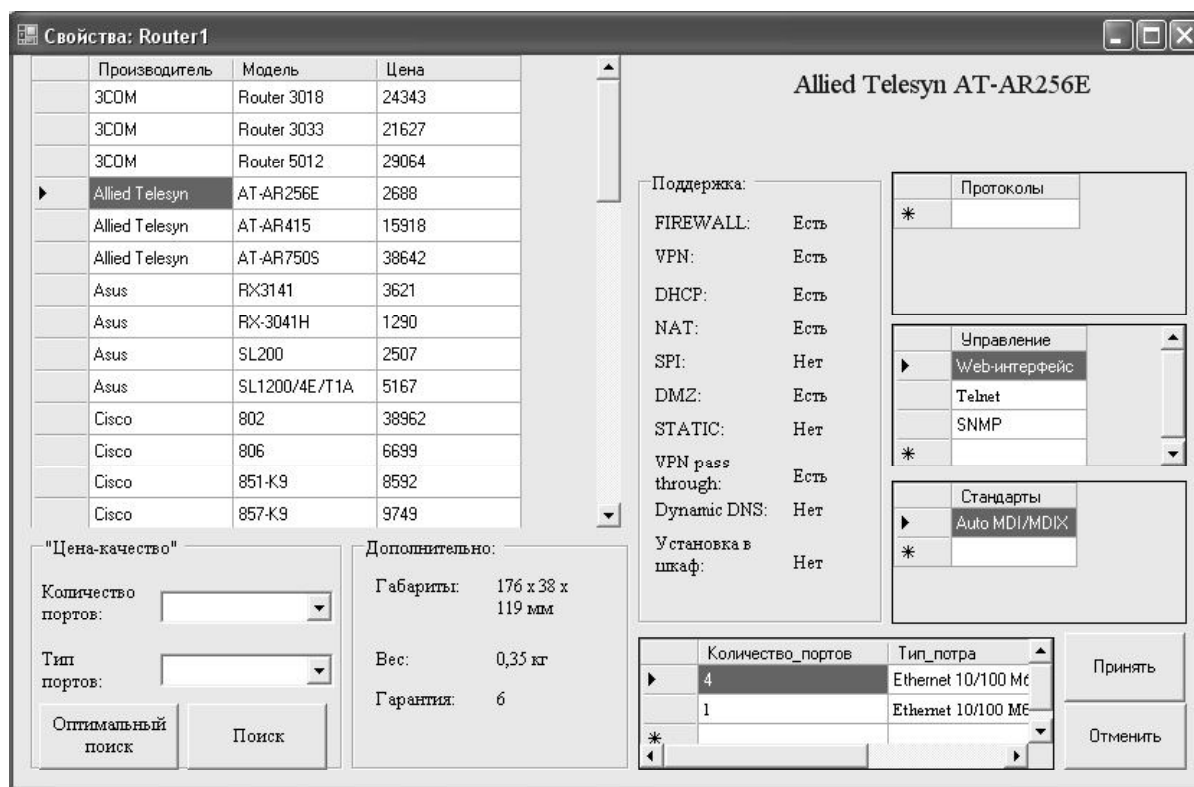


Рис. 3. Вид окна для выбора маршрутизатора

По нажатию кнопки «Поиск» формируется полный список имеющегося в БД оборудования с указанными характеристиками. Нечёткий выбор оборудования инициируется при нажатии кнопки «Оптимальный поиск».

В ходе экспериментов проводился нечёткий поиск в таблицах, содержащих 1500, 3000 и 5000 записей. В результате экспериментов установлено, что отказ от хранения в таблице значений функций принадлежности восьми термов двух ЛП позволил сократить объём хранимых данных на 33,6 %. Кроме того, исчезла необходимость в разработке средств вычисления указанных значений при обновлении и добавлении записей в таблицу. Время выполнения SQL-запросов, вычисляющих значения функций принадлежности на лету, и запросов, осуществляющих выборку этих значений из таблицы, практически одинаково (менее 1 с) и незначительно изменялось с увеличением объёма данных (0,95–0,98 с и 0,89–0,94 с соответственно).

**Заключение.** Представленная система содержит средства организации нечёткого поиска в БД на основе качественных критериев и тем самым существенно упрощает пользователю выбор оборудования, наилучшим образом удовлетворяющего его потребностям и возможностям. Выбор выполняется с помощью SQL-запросов, реализующих нечёткий поиск и встроенных в интерфейс системы. Предложенный алгоритм формирования SQL-запросов применим для реализации нечёткого поиска в различных предметных областях, позволяет использовать возможности СУБД для выполнения необходимых вычислений, не требует от пользователя дополнительных знаний в облас-

ти нечёткой логики, автоматизирует процесс формирования функций принадлежности термов лингвистической переменной. Направление дальнейших работ будет связано с расширением возможностей интерфейса системы с целью предоставления пользователю средств непосредственного конструирования критериев нечёткого поиска, выбора нечётких модификаторов, формирования лингвистических переменных, а также организации хранения этой информации в базе данных.

### **Библиографический список**

1. Штовба, С. Д. Введение в теорию нечётких множеств и нечёткую логику / С. Д. Штовба. — Режим доступа: <http://matlab.exponenta.ru/fuzzylogic/book1/> (дата обращения : 14.10.2013).
2. Андрейчиков, А. В. Интеллектуальные информационные системы / А. В. Андрейчиков. — Москва : Финансы и статистика, 2006. — 423 с.
3. Смагин, С. С. Интеллектуальные информационные системы / А. А. Смагин, С. В. Липатова, А. С. Мельниченко. — Ульяновск : Ульян. гос. ун-та, 2010. — 136 с.
4. Zadeh, L. A. Fuzzy sets. Information and Control. 1965, vol. 8, iss. 3, pp. 338–353.
5. Базы данных. Интеллектуальная обработка информации / В. В. Корнеев [и др.] — Москва : Изд. Молгачева С. В. : Нолидж, 2001. — 496 с.
6. Паклин, Н. Б. Нечёткие запросы к реляционным базам данных / Н. Б. Паклин. — Режим доступа: <http://www.basegroup.ru/> (дата обращения : 14.10.2013).
7. Чуйкова, Е. Н. Формирование нечётких запросов к базе данных. / Е. Н. Чуйкова, А. А. Молчанов // Сб. конкурс. работ Всерос. смотра-конкурса науч.-техн. творчества студентов высш. учеб. заведений «Эврика-2007». — Новочеркасск : Оникс+, 2007. — С. 38–40.
8. Моисеенко, С. И. Трансляция операторов реляционной алгебры в SQL-операторы / С. И. Моисеенко, В. А. Левченко // Вестн. Дон. гос. техн. ун-та — 2005. — Т. 1, № 1. — С. 691–696.
9. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning — I. Information Sciences. 1975, vol. 8, iss. 3, pp. 199–249.
10. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning — II. Information Sciences. 1975, vol. 8, iss. 4, pp. 301–357.
11. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning — III. Information Sciences. 1975, vol. 9, iss. 1, pp. 43–80.
12. Программная система для ввода экспертных знаний / В. П. Димитров [и др.] // Вестн. Дон. гос. техн. ун-та. — 2011. — Т. 11, № 1. — С. 83–90.
13. Бекаревич, Ю. Б. Самоучитель Access 2009 / Ю. Б. Бекаревич, Н. В. Пушкина. — Санкт-Петербург : БХВ-Петербург, 2009. — 720 с.

Материал поступил в редакцию 06.11.2013.

### **References**

1. Shtovba, S. D. Vvedeniye v teoriyu nechetkikh mnozhestv i nechetkuyu logiku. [Introduction to the theory of fuzzy sets and fuzzy logic.] Available at: <http://matlab.exponenta.ru/fuzzylogic/book1/> (accessed: 14.10.2013) (in Russian).
2. Andreychikov, A. V. Intellekturnyye informatsionnyye sistemy. [Intelligent Information Systems.] Moscow: Finansy i statistika, 2006, 423 p. (in Russian).
3. Smagin, S. S., Lipatova, S. V., Melnichenko, A. S. Intellekturnyye informatsionnyye sistemy. [Intelligent Information Systems.] Ulyanovsk : UIGU, 2010, 136 p. (in Russian).
4. Zadeh, L. A. Fuzzy sets. Information and Control, 1965, vol. 8, iss. 3, pp. 338–353.

5. Korneyev, V. V., et al. Bazy dannykh. Intellekturnaya obrabotka informatsii. [Database. Intelligent information processing.] Moscow : Publ. Molgacheva S. V., Knowledge, 2001, 496 p. (in Russian).
6. Paklin, N. B. Nechetkiye zaprosy k relyatsionnym bazam dannykh. [Fuzzy demands to relational databases.] Available at: <http://www.basegroup.ru/> (accessed : 14.10.2013) (in Russian).
7. Chuykova, E. N., Molchanov, A. A. Formirovaniye nechetkikh zaprosov k baze dannykh. [Assignment of fuzzy demands to the database.] Sbornik konkursnykh rabot Vserossiyskogo smotra-konkursa nauchno-tehnicheskogo tvorchestva studentov vysshih uchebnykh zavedeniy «Evrika-2007». [Collection of entries of the All-Russian competition of sci and tech creativity of university students "Eureka-2007".] Novocherkassk : Oniks+, 2007, pp. 38–40 (in Russian).
8. Moiseyenko, S. I., Levchenko, V. A. Translyatsiya operatorov relyatsionnoy algebry v SQL-operatoru. [Translation of relation algebra statements to SQL – statements.] Vestnik of DSTU, 2005, vol. 1, no. 1, pp. 691–696 (in Russian).
9. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning — I. Information Sciences, 1975, vol. 8, iss. 3, pp. 199–249.
10. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning — II. Information Sciences, 1975, vol. 8, iss. 4, pp. 301–357.
11. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning — III. Information Sciences, 1975, vol. 9, iss. 1, pp. 43–80.
12. Dimitrov, V. P., et al. Programmnaya sistema dlya vvoda ekspertnykh znaniy. [Programmed system for input expert knowledge.] Vestnik of DSTU, 2011, vol. 11, no. 1, pp. 83–90 (in Russian).
13. Bekarevich, Y. B., Pushkina, N. V. Samouchitel Access 2009. [Access 2009 know-how book.] Sankt-Peterburg : BKhV-Peterburg, 2009, 720 p. (in Russian).

## **IMPLEMENTATION OF FUZZY SELECTION OF EQUIPMENT IN INFORMATION NETWORK DESIGN SYSTEM\***

**E. N. Chuykova**

*The implementation of the equipment selection process in the information network design system on the basis of fuzzy criteria is considered. The selection is made in the relational database by fuzzy SQL-queries generated in the design system program shell. The algorithm of the fuzzy SQL-queries assignment is proposed. An example of constructing a fuzzy query to select the best equipment in terms of price and quality is given. A fragment of the graphic system interface which contains an equipment selection window is presented. The described method of performing fuzzy queries is applicable to the implementation of a fuzzy search in various subject areas, it does not require any additional knowledge in fuzzy logic, and it automates the membership function formation process of the linguistic variable terms. The future activities will be linked to the expansion of the system interface capabilities on the direct designing of the fuzzy queries by a user.*

**Keywords:** relational database, SQL-queries, fuzzy logic, linguistic variable, fuzzy set, membership function.

---

\* The research is done within the frame of the independent R&D.