



УДК 004.8

**ПОДХОД К ХРАНЕНИЮ БАЗ НЕЧЕТКИХ ЗНАНИЙ**

Глоба Л. С., Терновой М. Ю., Штогрин Е. С.

*Национальный технический университет Украины «Киевский политехнический институт»,  
Институт телекоммуникационных систем,**г. Киев, Украина***lgloba@its.kpi.ua****ternovoy@its.kpi.ua****L\_Shtogrina@mail.ru**

В работе предложен подход к хранению баз нечетких знаний в реляционной базе данных. Приведена модель базы нечетких знаний как объединение частичных баз. Описаны этапы проектирования схемы базы данных для хранения базы нечетких знаний.

**Ключевые слова:** база нечетких знаний, лингвистические переменные, отношение, реляционная база данных.

**ВВЕДЕНИЕ**

Повышение требований к обработке информации в различных областях жизнедеятельности человека требует улучшения существующих и создания новых подходов, позволяющих более эффективно хранить, обрабатывать и представлять информацию в процессе принятия различных решений. Среди существующих подходов к обработке информации следует отдельно выделить те, которые базируются на использовании баз знаний (БЗ) [Субботин, 2008]. Данные подходы позволяют объединить знания экспертов и в последующем их использовать в автоматизированном режиме. При этом необходимо учитывать, что эксперт не всегда оперирует четкой информацией, а также использует в своих рассуждениях не только количественные, но и качественные категории. Для хранения таких знаний используются базы нечетких знаний (БНЗ) [Ротштейн, 1999].

Существует два основных подхода к представлению знаний: процедурный и декларативный [Братко, 2004; Субботин, 2008]. При использовании процедурного подхода знания представляются в виде программного кода, который выполняется. При декларативном – в виде фактов, с которыми работает исполняемая программа. Второй подход является более эффективным для использования в распределенной информационно-телекоммуникационной среде, так как позволяет одновременно использовать базу знаний различным программам.

При декларативном подходе существуют

различные форматы хранения знаний, такие как текстовые файлы, xml документы [Zhang et al. 2011], специальные языки представления знаний [KML].

Представление сложных баз знаний в указанном выше виде затрудняет их одновременное использование. Также недостатком такого хранения является сложность поддержания целостности и непротиворечивости, дублирование знаний, а также отсутствие универсального механизма доступа к знаниям. Выходом из данной ситуации может стать использование реляционных баз данных для хранения БЗ, которые, как раз, и ориентированы на работу в распределенной информационно-телекоммуникационной среде и позволят повысить эффективность работы со знаниями.

**1. Модель базы нечетких знаний**

В данной работе рассматривается представление знаний в виде нечеткой логической модели. В такой модели выделяются три части: язык, система аксиом и правила вывода. В случае модели сформированной средствами нечеткой логики язык представляет собой лингвистические переменные (ЛП), их терм-множества и структуру правил, к аксиомам относится база нечетких знаний (БНЗ), а правила вывода задаются формулами, с помощью которых определяется значение функции принадлежности (ФП) ЛП и имеют вид:

$$\text{ЕСЛИ}(X_{ml} = a_{ml}) \text{И} \\ (X_{jl} = a_{jl}) \text{И} \dots \text{И}(X_{kl} = a_{kl}) \text{ТО}(Y = d_{nl})'$$

где  $X_{ml}, X_{jl}, X_{kl}, Y$  - лингвистические переменные, которые оцениваются качественными термами  $a_{ml}, a_{jl}, a_{kl}, d_{nl}$  соответственно,  $l$  обозначает номер правила.

Одним из недостатков данной модели является то, что в общем случае БНЗ неупорядоченная, что приводит к полному перебору правил при логическом выводе. Избежать данного недостатка можно используя подход сведения БНЗ к древовидной структуре, который рассматривается в работе [Глоба и др., 2008]. Также сведение БНЗ к древовидной структуре дает возможность отобразить природную иерархичность корпоративных систем, что позволяет наглядно проследить зависимости и связи между объектами. После применения данного подхода БНЗ будет состоять из частичных БНЗ, которые в совокупности организуют древовидную структуру. Обозначим частичную БНЗ как  $\hat{R}$ . Каждая из частичных БНЗ представляется в виде матрицы знаний, and/or графа или правил вида:

$$\begin{aligned} & \text{ЕСЛИ}(X_1 = a_1^{k1}) \text{ И } (X_2 = a_2^{k1}) \text{ И... И } (X_S = a_S^{k1}) \text{ ИЛИ} \\ & (X_1 = a_1^{k2}) \text{ И } (X_2 = a_2^{k2}) \text{ И... И } (X_S = a_S^{k2}) \text{ ИЛИ..} \\ & \text{ИЛИ} (X_1 = a_1^{kq}) \text{ И } (X_2 = a_2^{kq}) \text{ И... И } (X_S = a_S^{kq}) \end{aligned}$$

$$\text{ТО } Y = d_k$$

где  $a_i^{kr}$  - нечеткий терм, которым оценивается ЛП  $X_i$ ;  $k = \overline{1, n}$ , где  $n$  - количество разных значений, которые используются для оценки выходной переменной  $Y$ ;  $r = \overline{1, q}$ , где  $q$  количество элементарных конъюнкций в данном правиле.

## 2. Схема базы данных для хранения базы нечетких знаний

Необходимость использования баз нечетких знаний различными приложениями в распределенной информационно-телекоммуникационной среде указывает на необходимость создания реляционной базы данных, что позволит эффективно хранить и использовать знания.

Для решения вышеописанной проблемы необходимо решить задачу формирования схемы реляционной БД для хранения представленной модели БНЗ. Эту задачу можно записать в виде:

Дано:

Математическая модель БНЗ:

1.  $\{X_i | i = \overline{1, v}\}$  - множество лингвистических переменных (ЛП), которым соответствуют  $\{x_i | i = \overline{1, v}\}$ , где ЛП - пятерка  $\langle X, T, U, G, H \rangle$ , в которой:

- 1.1.  $X$  - имя ЛП;
  - 1.2.  $T = \{t_j | j = \overline{1, z}\}$ , где  $t_j$  - терм-множество ЛП  $X$  с соответствующими функциями принадлежности (ФП)  
 $M = \{\mu_{X_j} | j = \overline{1, z}\}$ .
  - 1.3.  $U$  - универсальное множество, область определения ЛП;
  - 1.4.  $G = \{g_k | k = \overline{1, s}\}$  - синтаксические правила, в виде грамматики, которые порождают названия термов;
  - 1.5.  $H = \{h_k | k = \overline{1, s}\}$  - семантические правила, которые задают функции принадлежности нечетких термов, порожденных синтаксическими правилами  $G$ .
  2.  $y$  - результирующая переменная;
  3.  $Y$  - ЛП, которая соответствует  $y$ ;
- $BZ = \bigcup \hat{R}$  - древовидная база нечетких знаний

(БНЗ), где  $\hat{R}$  - частичная БНЗ [Глоба и др., 2008].

Найти:

Схему реляционной БД  $\langle B, L, C \rangle$  для хранения БНЗ, где

1.  $B = \{b_q | q = \overline{1, n}\}$  - множество отношений БД, для которых определены атрибуты  $A_q = \{a_{qp} | p = \overline{1, m}\}$  и накладываемые на них ограничения  $S_p = \{s_{pk} | k = \overline{1, r}\}$ ;
2.  $L = \{l_v | v = \overline{1, h}\}$  - множество связей между отношениями  $B$ .
3.  $C$  - правила и ограничения, которые обеспечивают целостность БД и поддерживают правильность внесения знаний.

Для решения поставленной задачи опишем концептуальный и логический этапы проектирования БД в соответствии с [Коннолли и др., 2003].

Первым шагом на этапе концептуального проектирования БД является определение сущностей предметной области. Исходя из описанной выше модели БНЗ, можно выделить две сущности, такие как ЛП "LinguisticVariable" и правило вывода "Rule". Связь между этими сущностями будет отображать, что ЛП является частью правила. Также для более детального описания можно ввести третью сущность ФП "MembershipFunction". Рассматривая информацию, которая описывает эти сущности можно определить свойства для этих сущностей. Для описания ЛП необходимо описать каждый ее терм, поэтому свойствами будут "название ЛП", "полное название термина ЛП" и "функция принадлежности". Для правила необходимо описать информацию о его

левой и правой частях, поэтому для сущности “Rule” свойствами будут “условие” и “следствие”. Также необходимо предусмотреть то, что правило содержит не одну составляющую “условие”, а несколько.

На следующем шаге концептуального проектирования необходимо определить ограничения, которые будут обеспечивать целостность данных и соблюдение особенностей информации представления нечетких знаний:

1. ЛП должна быть задана на множестве ее определения.
2. Если ЛП содержится в условии заданного правила, то она не может быть следствием этого правила.
3. Если два правила имеют одинаковые условия, то они не могут иметь различные следствия.

Далее рассмотрим этап логического проектирования. Так как выбран реляционный тип БД, то просто сопоставить сущности с отношениями, а свойства с атрибутами отношений нельзя, так как полученная в результате такого сопоставления схема будет не нормализована, в ней будут присутствовать аномалии и избыточность данных. Также необходимо учесть, что разные ЛП могут иметь одинаковые названия термов и синтаксические правила, которые образуют названия термов. Следовательно, для уменьшения избыточности можно определить классификаторы для таких множеств, которые возможно будет расширять и дополнять. Зададим их с помощью введения отношений “TermName” и “RuleForCreateTermName” соответственно. В результате появляется связь многие-ко-многим (каждой ЛП соответствует несколько термов, и каждый терм может соответствовать нескольким ЛП), которую в реляционной БД необходимо реализовать с помощью третьего отношения, в данном случае “Term”.

Так как любую непрерывную функцию можно аппроксимировать сколь угодно близко кусочно-линейной функцией, то можно допустить, что функции принадлежности задаются не в виде формул, а в виде наборов координат. Такое допущение позволит хранить информацию о ЛП, правилах и ФП в одном месте, и не зависеть от программного обеспечения, которое реализует определение и вычисление функций принадлежности. Еще одним преимуществом такого хранения является возможность введения классификатора для ФП (реализован отношением “MembershipFunction”), что в дальнейшем предоставит возможность определения имеют ли различные ЛП одинаковые ФП. Для хранения координат ФП введем отношение “Func\_XY”.

Для хранения правил определим отношения “FuzzyRule” и “KnowledgeMatrix”.

Предложенная схема хранения будет состоять из отношений  $B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$ , где

- $b_1$  - отношение “LinguisticVariable” в котором  $A_1 = \{a_{11}, a_{12}, a_{13}\}$ , где  $a_{11}$  - “LinguisticVariableID” уникальный идентификатор ЛП,  $a_{12}$  - “Name” название ЛП,  $a_{13}$  - “Description” описание ЛП.
- $b_2$  - отношение “RuleForCreateTermName” в котором  $A_2 = \{a_{21}, a_{22}\}$ , где  $a_{21}$  - “RuleForCreateTermNameID” уникальный идентификатор,  $a_{22}$  - “Name” синтаксическое правило для порождения названия термина.
- $b_3$  - отношение “TermName”, в котором  $A_3 = \{a_{31}, a_{32}\}$ , где  $a_{31}$  - “TermID” уникальный идентификатор,  $a_{32}$  - Name название термина.
- $b_4$  - отношение “MembershipFunction”, в котором  $A_4 = \{a_{41}, a_{42}, a_{43}\}$ , где  $a_{41}$  - “MembershipFunctionID” уникальный идентификатор,  $a_{42}$  - “Name” название функции,  $a_{43}$  - “Description” описание функции.
- $b_5$  - отношение “Func\_XY”, в котором  $A_5 = \{a_{51}, a_{52}, a_{53}\}$ , где  $a_{51}$  - “MembershipFunctionID” уникальный идентификатор,  $a_{52}$  - “X” X координата функции,  $a_{53}$  - “Y” Y координата функции.
- $b_6$  - отношение “Term”, в котором  $A_6 = \{a_{61}, a_{62}, a_{63}, a_{64}\}$ , где  $a_{61}$  - “LinguisticVariableID”,  $a_{62}$  - “TermID”,  $a_{63}$  - “RuleForCreateTermNameID”,  $a_{64}$  - “MembershipFunctionID”.  $a_{61}, a_{62}, a_{63}, a_{64}$  - образуют составной ключ. Отношение “Term” является связующим звеном для всех остальных отношений.
- $b_7$  - отношение “FuzzyRule”, в котором  $A_7 = \{a_{71}, a_{72}\}$ , где  $a_{71}$  - “FuzzyRuleID” уникальный идентификатор,  $a_{72}$  - “Description” описание правила.
- $b_8$  - отношение “KnowledgeMatrix”, в котором  $A_8 = \{a_{81}, a_{82}, a_{83}, a_{84}, a_{85}\}$ , где  $a_{81}$  - “FuzzyRuleID”,  $a_{82}$  - “LinguisticVariableID”,  $a_{83}$  - “TermID”,  $a_{84}$  - “RuleForCreateTermNameID”,  $a_{85}$  - “IsResultOfRule” признак того, что это следствие правила.  $a_{81}, a_{82}, a_{83}, a_{84}$  - образуют составной ключ.

Для описания связей приведем графическое представление предлагаемой схемы БД (Рис. 1).

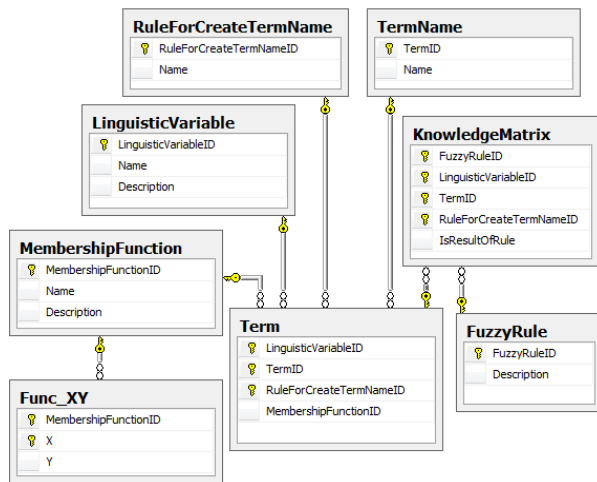


Рисунок 1 - Схема БД для хранения БНЗ

Правила и ограничения, которые обеспечивают целостность БД и описывают особенности представления знаний в БНЗ, определенные на этапе концептуального проектирования и описанные выше, могут быть заданы в виде триггеров в СУБД, которые будут срабатывать при добавлении и изменении информации.

## ЗАКЛЮЧЕНИЕ

Предложена схема хранения БНЗ в реляционной БД. Данный подход позволяет использовать все преимущества хранения, и работы с информацией предоставляемые реляционными БД, при этом работать со знаниями которые предоставляются в виде БНЗ. За счет хранения знаний в реляционной модели можно извлекать только необходимые знания для построения логического вывода. Повысить эффективность получения знаний, используя предложенную схему БД возможно введением индексации, что позволит ускорить выполнение запросов.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Глоба и др., 2008] Глоба, Л. С. Створення баз нечітких знань для інтелектуальних систем управління / Л. С. Глоба, М. Ю. Терновий, О. С. Штогріна // Комп'ютеринг. - Міжнародний науково-технічний журнал. - том 7, випуск 1. - Тернопіль, «Економічна думка» - 2008. - С.70-79.
- [Ротштейн, 1999] Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети / А. П. Ротштейн // Винница: УНИВЕРСУМ - 1999. - 320 с.
- [Коннолли и др., 2003] Коннолли Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг, А. Страчан // М.: Издательский дом «Вильямс», 3-е изд.: Пер. с англ.: Уч. пос. 2003.- 1440с.
- [Братко, 2004] Братко И. Алгоритмы искусственного интеллекта на языке PROLOG / И. Братко // М.: Издательский дом «Вильямс» - 2004. - 640 с.
- [Субботин, 2008] Субботин С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / С. О. Субботін // Навчальний посібник. — Запоріжжя: ЗНТУ, 2008. — 341 с.
- [Zhang et al., 2011] Zhang X. A knowledge-based approach for answering fuzzy queries in XML / X. Zhang, X. Meng, X. Wang // Seventh International Conference on Natural Computation (ICNC). - 2011. - pp. 18-22.

[KML] KML (Knowledge Management Tools) [Электронный ресурс]. - Электрон. текстовые данные. - Режим доступа: <http://kml.mipt.ru/A/ru/bin/view/Home/KML2Specification>

## APPROACH FOR STORING FUZZY KNOWLEDGE BASES

Globa L.S., Ternovoy M.Y., Shtogrina O.S.

National Technical University of Ukraine "Kyiv Polytechnic Institute", Kiev, Ukraine

lgloba@its.kpi.ua

ternovoy@its.kpi.ua

L\_Shtogrina@mail.ru

The paper proposes the approach for fuzzy knowledgebase storing in the relational database. The design process of database scheme is described.

## INTRODUCTION

Based on knowledgebase approaches allow to integrate expert knowledge and use them in computer-aided mode. Fuzzy knowledgebases can be highlighted from another knowledgebases as type of knowledgebases which contain fuzzy rules obtained from expert. There are some formats for knowledgebase storing such as text files, xml-based documents etc. The disadvantages of mentioned formats are complexity of consistency maintenance and absence of data access unified mechanism. The solution of the described problem is to use relational databases for knowledgebase storing.

## MAIN PART

Fuzzy knowledgebase can be described as union of partial fuzzy knowledgebases each of which contains rules which defined one linguistic variable in the form IF-AND-OR-THEN.

The problem statement can be defined as follows: to obtain relational database scheme if we know the mathematical model of fuzzy knowledgebase.

The design process consists of conceptual and logical steps. At the end of this process relational database scheme is obtained (Fig. 1). There are also defined several constraints which can help to maintain integrity and specific features of fuzzy knowledge storing.

## CONCLUSION

Proposed approach allows to increase efficiency of using fuzzy knowledge in information-telecommunication environment because of using unified methods to store and retrieve information which are given by the relational databases. The velocity of knowledge retrieval can be additionally increased by using special type of indexing.