

## Інформаційні системи і технології

УДК 004.9

### Метод збереження та використання баз нечітких знань

**М.Ю.Терновой**, канд. техн. наук, **О.С.Штогріна**

Національний технічний університет України "Київський політехнічний інститут",  
Інститут телекомунікаційних систем,  
провулок Індустріальний, 2, Київ-56, 03056, Україна.

**В роботі запропонований метод зберігання баз нечітких знань в реляційній базі даних. Описана схема бази даних для зберігання нечітких знань. Описані основні етапи побудови дерева нечіткого логічного виведення шляхом вибору необхідних правил з бази даних. Бібл.17, рис. 1.**

**Ключові слова:** база нечітких знань, лінгвістичні змінні, відношення, реляційна база даних.

#### Вступ

В сучасних умовах успішна робота підприємств і організацій залежить від можливостей інформаційно-телекомунікаційних систем, які дозволяють проводити збір, зберігання та обробку інформації, а також надають підтримку прийняття рішень на основі цієї інформації. Важливим напрямком при розробці таких систем є використання підходів, що дозволяють використовувати експертні знання [11, 12]. Такі підходи передбачають збір, постійне накопичення нових та довготривале зберігання знань багатьох експертів в базах знань (БЗ), що в подальшому дозволить використовувати ці знання в автоматизованому режимі.

Як правило, експерти подають свої знання у вигляді правил «Якщо-То» [12]. Необхідно приймати до уваги те, що експерт не завжди оперує чіткою інформацією, а також використовує в своїх міркуваннях не тільки кількісні, але і якісні категорії. Для зберігання таких знань призначені бази нечітких знань (БНЗ) [11, 15].

Існує два основних підходи до подання знань: процедурний і декларативний [10, 12, 16]. При використанні процедурного підходу знання представляються у вигляді програмного коду, який виконується. Цей підхід передбачає залучення програміста на всіх етапах внесення та корегування знань у БЗ. При декларативному підході знання подаються у вигляді фактів, з якими працює програма, що їх інтерпретує. Іс -

нують різні формати зберігання фактів, такі як текстові файли, xml документи [9], спеціальні мови представлення знань [5]. Недоліком такого зберігання є складність підтримки цілісності і несуперечності, можливе дублювання знань, а також відсутність універсального механізму доступу до знань.

Для усунення цих недоліків в роботах [2,3,6,13,17] пропонується використовувати реляційні бази даних (БД) для зберігання нечітких знань. Реляційні БД, на відміну від текстових та xml документів, забезпечують підтримку цілісності відповідно заданим вимогам та орієнтовані на роботу в розподіленому інформаційно-телекомунікаційному середовищі.

Дослідження, які проводяться в напрямку використання реляційної моделі для збереження нечітких знань, спрямовані на задачі збереження нечітких, невизначених, неповних даних. У роботах [2,3,13,17] розглядаються випадки розширення ER моделі для зберігання нечітких даних за рахунок введення нечітких атрибутів, нечітких сутностей, нечітких зв'язків та ін. В багатьох випадках запропоновані розширення ER моделі є проблемно-орієнтованими [6,13,17]. В існуючих роботах недостатня увага приділяється збереженню власне баз нечітких знань, які на ряду з інформацією про лінгвістичні змінні (ЛЗ) та їх терм-множини зберігають правила, які описують їх взаємозалежність. Окремим напрямом є розробка та використання діалектів мови SQL [1,7,8], для постановки запитів на отримання та маніпулювання нечіткими даними.

Виходячи з вищезазначеного поставлена в даній роботі задача розробки підходу до використання реляційних БД в якості сховища БНЗ є актуальною. Даний підхід дозволить підвищити ефективність роботи зі знаннями, а також в подальшому розробити методи об'єднання БНЗ для використання інтегрованих знань у розподіленому інформаційно-телекомунікаційному середовищі підприємства.

**Схема бази даних для зберігання бази нечітких знань**

У даній роботі розглядається подання знань у вигляді нечіткої логічної моделі. У такій моделі виділяються три частини: мова, система аксіом і правила виведення. У випадку моделі сформованої засобами нечіткої логіки мова являє собою лінгвістичні змінні (ЛЗ), їх терм-множини і структуру правил, до аксіом відноситься база нечітких знань, а правила виведення мають вигляд «Якщо-То».

Постановку задачі формування схеми реляційної БД для зберігання БНЗ можна записати у вигляді:

Дано:

1.  $\{X_i \mid i = \overline{1, N_X}\}$  - множина лінгвістичних змінних, де  $N_X$  - кількість ЛЗ, а ЛЗ - п'ятірка  $\langle X, T, U, G, H \rangle$ , в якій:

1.1.  $X$  - ім'я ЛЗ;

1.2.  $T = \{t_k \mid k = \overline{1, N}\}$  - терм-множина лінгвістичної змінної  $X$  (множина допустимих значень ЛЗ  $X$ ) з відповідними функціями належностей (ФН)  $M = \{\mu_{t_k} \mid k = \overline{1, N}\}$ , де  $t_k$  - терм, представляється як нечітка множина на універсальній множині  $U$ ,  $k$  - номер терму,  $N$  - кількість термів ЛЗ  $X$ ;

1.3.  $U$  - універсальна множина, область визначення ЛЗ;

1.4.  $G = \{g_p \mid p = \overline{1, s}\}$  - синтаксичні правила, у вигляді граматики, які породжують назви термів,  $s$  - кількість синтаксичних правил;

1.5.  $H = \{h_p \mid p = \overline{1, s}\}$  - семантичні правила, які задають функції належності нечітких термів, породжених синтаксичними правилами  $G$ .

$\{P_j \mid j = \overline{1, N_P}\}$  - множина правил, де  $N_P$  - кількість правил, та правило має вигляд:

$$P_j = \text{ЯКЩО} (X_{i_1} = t_{k_1}) \text{ТА} (X_{i_2} = t_{k_2}) \text{ТА} \dots, \\ \dots \text{ТА} (X_{i_n} = t_{k_n}) \text{ТО} (X_l = t_z)$$

де  $P_j$  -  $j$ -е правило для визначення  $z$ -го терму ЛЗ з ідентифікатором  $l$ ;

$j = \overline{1, M}$ , де  $M$  - кількість правил БНЗ;

$X_{i_s}$  - лінгвістична змінна, яка оцінюється якісним термом  $t_{k_s}$ ;

$s = \overline{1, n_j}$  - номер ЛЗ в лівій частині  $j$ -го правила,

$n_j$  - кількість ЛЗ, що знаходяться в лівій частині  $j$ -го правила,

$k_s = \overline{1, N_{i_s}}$  - номер терму яким оцінюється ЛЗ  $X_{i_s}$ ,

$N_{sj}$  - кількість термів ЛЗ  $X_{i_s}$ .

Зайти:

Схему реляційної БД  $\langle B, L, C \rangle$  для зберігання БНЗ, де:

1.  $B = \{b_q \mid q = \overline{1, n}\}$  - множина відношень БД, для яких визначені атрибути  $A_q = \{a_{qp} \mid p = \overline{1, m}\}$  та обмеження  $S_p = \{s_{pk} \mid k = \overline{1, r}\}$ , що накладаються на ці атрибути;

2.  $L = \{l_v \mid v = \overline{1, h}\}$  - множина зв'язків між відношеннями  $B$ .

3.  $C$  - правила та обмеження, які забезпечують цілісність та підтримують правильність внесення знань.

Вирішення цієї задачі полягає у проведенні концептуального та логічного етапу проектування БД [14]. Спочатку виділяються основні сутності їх властивості та зв'язки між сутностями, потім для усунення аномалій та збитковості даних проводиться нормалізація отриманої схеми. Після проведення зазначених дій запропонована схема зберігання буде складатися з наступних відношень:  $B = \{LinguisticVariable, TermName, RuleForCreateTermName, MembershipFunction, FuncXY, Function, Term, KnowledgeMatrix, FuzzyRule\}$ , де

- Відношення `LinguisticVariable` призначено для зберігання лінгвістичних змінних, включає атрибути `LinguisticVariableID` - унікальний ідентифікатор ЛЗ, `Name` - ім'я ЛЗ та `Description` - опис ЛЗ.
- Відношення `TermName` використовується як класифікатор назв термів, включає атрибути `TermID` - унікальний ідентифікатор терму та `Name` - назву терму відповідно.
- Відношення `RuleForCreateTermName` використовується як класифікатор синтаксичних правил для формування назви терму, включає атрибути `RuleForCreateTermNameID` - унікальний ідентифікатор синтаксичного правила для формування назви терму та `Name` - ім'я синтаксичного правила для формування назви терму.
- Відношення `MembershipFunction` використовується як класифікатор для функцій належності, включає атрибути `MembershipFunctionID` - унікальний ідентифікатор функції належності, `FunctionID` - унікальний ідентифікатор функції, `Name` - ім'я, `Description` - опис, `Parameters` - параметри функції, в разі виклику функції з відповідної бібліотеки, `FromX`

- та ToX - границі області визначення функції належності.
- Відношення Function використовується для зберігання інформації для підключення до бібліотек, які реалізують обчислення функцій належності, включає атрибути FunctionID - унікальний ідентифікатор, Name - назву функції, ConnectionString - параметри, які описують де знаходиться та як викликати функцію та Description - опис функції.
  - Відношення FuncXY використовується як альтернативне до відношення Function, в тому випадку коли ФН можливо задати набором координат, наприклад, для трикутних або трапецієвидних ФН. В цьому відношенні зберігаються координати точок, по яким можна відтворити функцію належності. Включає атрибути, MembershipFunctionID - унікальний ідентифікатор функції належності, X - x координата точки та Y - y координата точки.
  - Відношення Term використовується для зберігання інформації про терми лінгвістичних змінних, об'єднує в собі всю інформацію з вищеписаних відношень, включає атри -

бути TermID - унікальний ідентифікатор терму, LinguisticVariableID - унікальний ідентифікатор лінгвістичної змінної, TermNameID - унікальний ідентифікатор назви терму, RuleForCreateTermName - унікальний ідентифікатор синтаксичного правила для формування назви терму та MembershipFunctionID - унікальний ідентифікатор функції належності.

- Відношення Rule використовується для зберігання інформації про правила БНЗ, включає атрибути FuzzyRuleID - унікальний ідентифікатор правила та Description - опис правила.
- Відношення KnowledgeMatrix використовується для зберігання власне правил БНЗ, включає атрибути FuzzyRuleID - унікальний ідентифікатор правила, TermID - унікальний ідентифікатор терму та IsResultOfRule - ознака, чи є цей терм результатом даного правила.

Для опису зв'язків, ключів відношень та наочного уявлення на рис. 1 наведено графічне представлення запропонованої схеми зберігання.

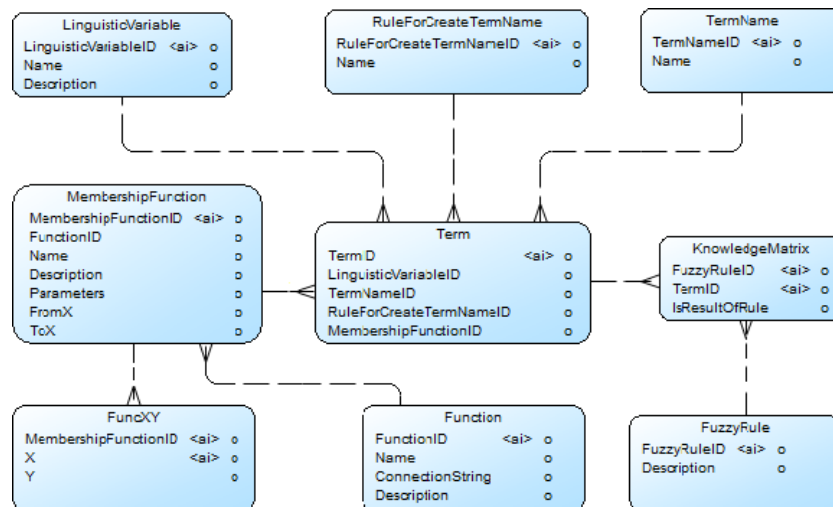


Рис. 1. Схема БД для зберігання БНЗ

Обмеження, які будуть забезпечувати цілісність і дотримання особливостей представлення нечітких знань можуть бути задані у вигляді доменів для атрибутів, власне забезпеченням цілісності засобами системи керування БД (СКБД) для запропонованої схеми, або тригерів в СКБД, які будуть спрацьовувати при додаванні та зміні інформації. Загальне формулювання цих обмежень наведено нижче:

- ЛЗ повинна бути задана на області її визначення;

- якщо ЛЗ міститься в умові заданого правила, то вона не може бути наслідком цього правила;
- якщо два правила мають однакові умови, то вони не можуть мати різні наслідки.

В запропонованій схемі кожна сутність (ЛЗ, терм, правило) ідентифікується за допомогою ключового атрибуту, який є унікальним. Це надає можливість оперувати сутностями та здійснювати логічне виведення, використовуючи тільки унікальні ідентифікатори сутностей, за рахунок цього скорочується кількість даних, що

передаються від БД до програми, що їх оброблює. Знаючи унікальний ідентифікатор сутності, можливо отримати всю інформацію про сутність. Додаткові дані, які описують сутності, що використовуються, вибираються та передаються по запиту користувача.

Отримання даних з БД описується за допомогою операцій реляційної алгебри [17], де  $\sigma_F(R)$  - операція вибірки кортежів, які відповідають предикату  $F$ , з відношення  $R$ .

$\Pi_A(R)$  - операція проєкції по атрибутам  $A$  над відношенням  $R$ .

$$\Pi_{Name, Description}(\sigma_{LinguisticVariableID=@LinguisticVariableID}(LinguisticVariable))$$

Інші операції по отриманню даних, що описують сутності здійснюються аналогічно.

### Побудова дерева виведення на основі бази нечітких знань, яка зберігається в реляційній базі даних

В загальному випадку нечітке логічне виведення складається з наступних етапів:

- Побудова дерева логічного виведення на основі правил БНЗ.
- Отримання значень вхідних лінгвістичних змінних та їх фазифікація.
- Здійснення нечіткого логічного виведення.
- Дефазифікація.

Запропонований метод зберігання нечітких знань у реляційній БД дозволяє зменшити обчислювальну складність проведення нечіткого логічного виведення за рахунок побудови дерева

$$\Pi_{FuzzyRuleID}(\sigma_{(TermID=@TermID)and(IsResultOfRule=true)}(KnowledgeMatrix))$$

2.2. Якщо для терму таких правил не існує, то терм є листом дерева та ступінь істинності цього терму отримується за рахунок операції фазифікації;

2.3. Пронумерувати терм та додати його до дерева нечіткого логічного виведення.

$$\Pi_{TermID}(\sigma_{(FuzzyRuleID=@FuzzyRuleID)and(IsResultOfRule=false)}(KnowledgeMatrix));$$

3.2. Сформувані правила, пронумерувати його та додати до дерева нечіткого логічного виведення.

3.3. Перехід на п. 2.

Отримання даних з БД для побудови дерева нечіткого логічного виведення можливо проводити рекурсивно або за рахунок використання такої структури даних, як черга. При рекурсивній процедурі процес побудови дерева закінчується коли на 2-гому кроці всі терми виявилися листами дерева, тоді не існує правил, які потрібно

Наприклад, отримання даних про ЛЗ, до якої належить терм по його унікальному ідентифікатору описується наступними операціями реляційної алгебри:

1. Отримати унікальний ідентифікатор ЛЗ

$$\Pi_{LinguisticVariableID}(\sigma_{TermID=@TermID}(Term));$$

2. Отримати дані про лінгвістичну змінну з ідентифікатором отриманим на попередньому кроці:

нечіткого логічного виведення для знаходження значення тільки необхідної ЛЗ.

Побудова дерева нечіткого логічного виведення здійснюється шляхом вибору з БД необхідних правил, та складається з наступних кроків:

1. Отримати всі терми лінгвістичної змінної, значення якої необхідно визначити. Реалізується операціями вибірки та проєкції виконаними над відношенням Term:

$$\Pi_{TermID}(\sigma_{LinguisticVariableID=@LinguisticVariableID}(Term)).$$

2. Для кожного отриманого на попередньому кроці терму:

2.1. Визначити всі правила, в яких заданий терм є результатом. Реалізується операціями вибірки та проєкції виконаними над відношенням KnowledgeMatrix:

3. Для кожного отриманого на попередньому кроці правила:

3.1. Визначити всі терми з його лівої частини. Реалізується операціями вибірки та проєкції виконаними над відношенням KnowledgeMatrix:

обробляти на 3-му кроці. При процедурі побудови дерева з використанням черги, черга використовується для зберігання термів, які необхідно обробити. Коли черга стає пустою, то дерево сформоване.

В процесі формування дерева також вводиться нумерація термів та правил, яка в подальшому дозволить застосовувати правила та обчислювати ступені істинності термів у відповідності до застосованої нумерації, що дозволить

скоротити час проведення нечіткого логічного виведення.

Для запропонованого методу нижче наведено алгоритм з використанням черги, який подано у вигляді псевдокоду.

```

BUILDING_INFERENCE_TREE (LZ)
Черга Q
Масив Terms
Масив Rules
Number ← 0
Terms ← Застосування формули (1) з параметром @LinguisticVariableID=LZ
for i ← 0 to length[Terms]
    Enqueue(Q, Terms[i])
    i ← i + 1
end for
while length[Q] > 0
    term ← Dequeue(Q)
    Додати до InferenceTree term пронумерований номером Number
    Number ← Number + 1
    Rules ← Застосування формули (2) з параметрами @TermID=term та @IsResultOfRule=1
    i ← 0
    while i < length[Rules]
        Terms ← Застосування формули (3) з параметрами @FuzzyRuleID=Rules[i] та @IsResultOfRule=0
        Додати до InferenceTree правило пронумероване номером Number, в якому результатом є term та лівою частиною є терми з Terms
        Number ← Number + 1
        for j ← 0 to length[Terms]
            if в Q не існує елемента Terms[j]
                then Enqueue(Q, Terms[j])
            j ← j+1
        end for
        i ← i+1
    end while
end while

```

## Висновки

Запропоновано схему зберігання БНЗ в реляційній БД та метод побудови дерева нечіткого логічного виведення на основі отримання правил з БД. Даний метод дозволяє використовувати всі переваги зберігання, і роботи з інформацією, які надаються реляційними БД. Також БНЗ одночасно може використовуватись різними програмами. Тиражування знань для використання в різних інформаційно-телекомунікаційних системах в цьому випадку здійснюється копіюванням БД, або наданням віддаленого доступу до існуючої БД, в якій зберігається БНЗ. Прис-

корити операції вибірки з БД можливо введеним індексації.

Запропоновану схему зберігання можливо розширити та використовувати для зберігання багатьох БНЗ, за рахунок цього забезпечити інтеграцію декількох БНЗ.

## Література

1. *Adnan Yazici, Roy George* Fuzzy Database Modeling Studies in Fuzziness and Soft Computing (vol. 26), Springer, 1999. – 234 с.
2. *José Galindo, Angélica Urrutia, Mario Piattini* Fuzzy Databases: Modeling, Design And Implementation / Idea Group Inc (IGI), 2006. – 320 с.
3. *José Galindo, Angélica Urrutia, Mario Piattini* Representation of Fuzzy Knowledge in Relational Databases / Proceedings. 15th International Workshop on Database and Expert Systems Applications, 2004.. - Pp. 917-921.
4. KML (Knowledge Management Tools) [Електронний ресурс]. – Електрон. текстові дані. - Режим доступу: <http://kml.mipt.ru/A/ru/bin/view/Home/KML2Specification>
5. *Li Yan, Z. M. Ma, Jian Liu* Fuzzy data modeling based on XML schema / Proceedings of the 2009 ACM symposium on Applied Computing. New York, NY, USA 2009. Pp. 1563 – 1567.
6. *Nauman Chaudhry, James Moynes, Elke A. Rundensteiner* Designing Databases with Fuzzy Data and Rules for Application to Discrete Control, University of Michigan / Computer Science and Engineering Division, Department of Electrical Engineering and Computer Science, 1994. – 21 с.
7. FSQ (A Fuzzy Query Language) [Електронний ресурс]. – Електрон. текстові дані. – Режим доступу: <http://www.lcc.uma.es/~ppgg/FSQ.html#Ref>
8. *Srdjan Skrbic, Milos Rackovic, Aleksandar Takaci* The PFSQ Query Execution Process / Novi Sad J. Math. Vol. 41, No. 2, 2011. – Pp. 161-179.
9. *Zhang X., Meng X., Wang X.* A knowledge-based approach for answering fuzzy queries in XML / Seventh International Conference on Natural Computation (ICNC). - 2011. - pp. 18-22.
10. *Братко И.* Алгоритмы искусственного интеллекта на языке PROLOG / М.: Издательский дом «Вильямс» - 2004. - 640 с.
11. *Глоба Л. С., Терновой М. Ю., Штогріна О. С.* Створення баз нечітких знань для інтелектуальних систем управління / Комп'ютинг. - Міжнародний науково-

- технічний журнал. – том 7, випуск 1. – Тернопіль, «Економічна думка» – 2008. – С.70-79.
12. Джексон П. Введение в экспертные системы / Издательский дом «Вильямс», 2001. – 624 с.
  13. Касаткина Н.В., Танянский С.С., Филатов В.А. Методы хранения и обработки нечетких данных в среде реляционных систем / ААЭКС, №2(24), Информационно-управляющие комплексы и системы, 2009, – Электрон. текстові дані. – Режим доступу: <http://aaecs.org/kasatkina-nv-tanyanskii-ss-filatov-va-metodi-hrneniya-i-obrabotki-nechetkih-dannih-v-srede-relyacionnih-sistem.html>
  14. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Издательский дом «Вильямс», 3-е изд.: Пер. с англ.: Уч.пос. - М., 2003.- 1440с.
  15. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети / Винница: УНИВЕРСУМ - 1999. – 320 с.
  16. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / Навчальний посібник. — Запоріжжя: ЗНТУ, 2008. — 341 с.
  17. Филатов В.А., Касаткина Н.В., Винокурова Е.А. Интеллектуальный анализ и визуализация нечетких данных на основе метода главных компонент / Вестник ХНТУ №2(38), 2010. – С. 154 – 158.

УДК 004.9

## Метод хранения и использования баз нечетких знаний

М.Ю. Терновой, канд. тех. наук; Е.С. Штогрин

Национальный технический университет Украины “Киевский политехнический институт”,  
Институт телекоммуникационных систем,  
проулок Індустріальний, 2, Київ-56, 03056, Україна.

В работе предложен метод хранения и использования баз нечетких знаний в реляционной базе данных. Выделены основные сущности базы нечетких знаний. Приведено подробное описание схемы базы данных для хранения нечетких знаний, полученной после уточнения выделенных сущностей и проведения нормализации. Описаны основные этапы построения дерева нечеткого логического вывода путем выбора необходимых правил из базы данных. Процедуры взаимодействия с базой данных приведены в виде операций реляционной алгебры. Для предложенного метода приведен алгоритм реализации с использованием очереди. Предложенная в работе схема хранения и метод построения дерева нечеткого логического вывода позволяют повысить эффективность работы со знаниями, используя преимущества хранения и работы с информацией, которые предоставляются реляционными базами данных. Библ.17, рис. 1.

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

## Method for fuzzy knowledge bases storing and processing

M.Y. Ternovoy, O.S. Shtogrina

National Technical University of Ukraine “Kyiv Polytechnic Institute”,  
Institute of Telecommunication Systems,  
Industrialnyy Al., 2, Kyiv-56, 03056, Ukraine.

The method for fuzzy knowledge base storing and processing in relational database is proposed in the paper. The main entities are singled out from fuzzy knowledge base. The database scheme for fuzzy knowledge storing is described in details. The main steps for building fuzzy inference tree based on selecting rules from database are described. The interaction procedures

with database are described with the help of relation algebraic operation. Based on queue algorithm for proposed method is described. Proposed scheme and approach for fuzzy inference construction allow increasing work efficiency with knowledge because of using advantages of relational databases. Reference 17, figures 1.

**Key words:** *fuzzy knowledge base, linguistic variable, relation, relational algebra.*

1. [Adnan Yazici, Roy George](#) Fuzzy Database Modeling / [Studies in Fuzziness and Soft Computing \(vol. 26\)](#), Springer, 1999. – 234 с.
2. [José Galindo, Angélica Urrutia, Mario Piattini](#) Fuzzy Databases: Modeling, Design And Implementation / Idea Group Inc (IGI), 2006. – 320 с.
3. [José Galindo, Angélica Urrutia, Mario Piattini](#) Representation of Fuzzy Knowledge in Relational Databases / Proceedings. 15th International Workshop on [Database and Expert Systems Applications, 2004..](#) - Pp. 917-921.
4. KML (Knowledge Management Tools) / <http://kml.mipt.ru/A/ru/bin/view/Home/KML2Specification>
5. [Li Yan, Z. M. Ma, Jian Liu](#) Fuzzy data modeling based on XML schema / Proceedings of the 2009 ACM symposium on Applied Computing. New York, NY, USA 2009. Pp. 1563 – 1567.
6. [Nauman Chaudhry, James Moyne, Elke A. Rundensteiner](#) Designing Databases with Fuzzy Data and Rules for Application to Discrete Control, University of Michigan / Computer Science and Engineering Division, Department of Electrical Engineering and Computer Science, 1994. – 21 с.
7. FSQL (A Fuzzy Query Language) / <http://www.lcc.uma.es/~ppgg/FSQL.html#Ref>
8. [Srdjan Skrbic, Milos Rackovic, Aleksandar Takaci](#) The PFSQL Query Execution Process / Novi Sad J. Math. Vol. 41, No. 2, 2011. – Pp. 161-179.
9. [Zhang X., Meng X., Wang X.](#) A knowledge-based approach for answering fuzzy queries in XML / Seventh International Conference on Natural Computation (ICNC). - 2011. - pp. 18-22.
10. [Bratko I.](#) Artificial intelligence algorithms in the language PROLOG // Moscow.: Williams Publishing House - 2004. - 640 p. (Rus)
11. [Globa. L.S., Ternovoy M.Y., Shtogrina O.S.](#) Fuzzy Knowledgebase Design for Intellectual Systems / International Scientific Journal of Computing. – Vol. 7, Issue 1. – Ternopil, “Naukova dumka” – 2008. – Pp.70-79. (Ukr)
12. [Jackson P.](#) Introduction to Expert Systems / Williams Publishing House, 2001. – 624 p. (Rus)
13. [Kasatkyna N.V., Tanianskiy S.S., Fylatov V.A.](#) Methods for storing and processing fuzzy data in relational systems / “ААЭКС”, №2(24), Informatsyonno-upravliaiushchye komplekxy i systemy, 2009, – <http://aaecs.org/kasatkina-nv-tanyanskii-ss-filatov-va-metodi-hraneniya-i-obrabotki-nechetkih-dannih-v-srede-relyacionnih-sistem.html> (Rus)
14. [Connolly T., Begg C.](#) Database Systems. A Practical Approach to Design, Implementation, and Management / Moscow.: Williams Publishing House, Third Edition. - 2003.- 1440 p. (Rus)
15. [Rotshtein A. P.](#) Intellectual Technologies of Identification: Fuzzy Sets, Genetic Algorithms, Neural Nets / Vinnitsa.: UNIVERSUM - 1999. – 320 p. (Rus)
16. [Subbotin S.O.](#) Knowledge presentation and processing in artificial intelligence and decision support systems / Study book. — Zaporizhzhya: ZNTU, 2008. — 341 p. (Ukr)
17. [Fylatov V.A., Kasatkyna N.V., Vynokurova E.A.](#) Intelligent analysis and visualization of fuzzy data based on principal component analysis / Vestnyk KhNTU №2(38), 2010. – С. 154 – 158. (Rus)

Робота виконувалась за рахунок бюджетних коштів, наданих як грант Президента України для підтримки наукових досліджень молодих учених на 2012 рік № GP/F44/082.

Поступила в редакцію 27 октября 2012 г.