вого игрока. Показано, что для их определения достаточно воспользоваться концепцией седловой точки в известном принципе оптимальности с использованием соответствующего правого неравенства.

Ключевые слова: антагонистическая игра, выпуклая игра, оптимальная стратегия, оптимальная вероятность.

Romanuke V. V.

METHOD OF DETERMINATION OF THE FIRST PLAYER OPTIMAL STRATEGIES IN A SUBCLASS OF THE NONSTRICTLY CONVEX ANTAGONISTIC GAMES

By the example of two nonstrictly convex antagonistic games, where the second player has the single optimal pure strategy, it has been asserted, that there exists a subclass of nonstrictly convex antagonistic games, in which by the known method there cannot be determined the optimal probabilities of selecting the essential pure strategies of the first player. It has been demonstrated that to determine them, it is sufficient to employ the saddle point concept in the known optimality principle by applying the corresponding right-side inequality.

Key words: antagonistic game, convex game, optimal strategy, optimal probability.

УДК 51.001.57+004.652.4+004.827

Шаховська Н. Б.

Канд. техн. наук, доцент Національного університету «Львівська політехніка»

ФОРМАЛІЗАЦІЯ ПРОСТОРУ ДАНИХ ЗА ДОПОМОГОЮ АЛГЕБРАЇЧНОЇ СИСТЕМИ

Проаналізовано проблеми опрацювання розрізнених даних. Побудовано формальну модель простору даних та уведено операції над ним.

Ключові слова: простір даних, сховище даних, база даних, алгебраїчна система, пошук даних, групування даних, інтелектуальний агент, джерело даних.

ПОСТАНОВКА ПРОБЛЕМИ В ЗАГАЛЬНОМУ ВИГЛЯДІ

У різних галузях науки спостерігається експоненційний ріст обсягів експериментальних даних. Складність використання таких даних виникає внаслідок їхньої природної різнорідності (зберігання у різних системах, призначення для різних задач, різні методи опрацювання та зберігання тощо). Розрив, який збільшується між джерелами даних і сервісами, приводить до необхідності пошуку нових шляхів організації рішення задач над множинними розподіленими колекціями даних і програм, які концентруються в спеціалізованих центрах даних і обчислювальних ресурсах.

Традиційно при рішенні певних задач фахівці використовують звичні для них джерела інформації і формулюють завдання з огляду на лише на такі джерела. Очевидна неповнота інформації, яку вдається охопити при такому підході. Безліч джерел даних і сервісів, що існують в Інтернеті, їхня розмаїтість викликають потребу в радикальній зміні такого традиційного підходу. Сутність цієї зміни полягає в тому, що задачі повинні формулюватися незалежно від існуючих джерел інформації, і лише після такого формулювання повинна здійснюватися ідентифікація релевантних завданню джерел, приведення їх до виду, необхідного для розв'язання задачі, інтеграція, ідентифікація сервісів, які дозволяють реалізувати окремі частини абстрактного процесу рішення завдання.

Для прийняття адекватних рішень у певній галузі необхідно, щоб дані, які надходять із різних джерел і використовуються для прийняття керівних рішень, задовольняли такі вимоги:

– були повними, несуперечливими та надходили вчасно;

– були інформативними, оскільки вони застосовуватимуться для прийняття рішень;

– були однакової структури, щоб мати можливість завантажити їх у єдине сховище даних та проаналізувати;

– зберігалися в однакових моделях даних та були незалежними від платформи розроблення, щоб мати можливість використання цих даних іншими засобами.

Сьогодні найгостріші проблеми керування інформацією виникають в організацій (наприклад, готелів, баз відпочинку, оздоровчих закладів, туристичних агентств), робота яких полягає в опрацюванні великої кількості різнотипних, взаємонезалежних джерел даних. Такий тип системи отримав назву *простір даних*. На відміну від систем інтеграції даних, що також пропонують загальноприйнятий доступ до різ-

норідних джерел даних, простори даних не припускають, що всі семантичні взаємозв'язки між джерелами відомі і вказані. Багато користувачів, які працюють з просторами даних, проводять дослідження даних, і немає єдиної схеми, за якою вони можуть створювати запити.

АНАЛІЗ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

На сьогодні немає жодної методики опрацювання даних, яка б задовольняла всі наведені вимоги до опрацювання даних, а отже, немає можливості аналізувати стан галузі загалом, використовуючи першоджерела інформації, а не визначені наперед статистичні звіти [1, 2]. Розроблені методи інтеграції даних спираються на джерела даних із наперед визначеними структурами, які мають відомі механізми погодження [3, 4], що є неприпустимим у разі прийняття керівного рішення по усій предметній області.

Простір даних розглядають як нову абстракцію керування даними [4]. Основоположником ідеї просторів даних був Алон Хелеві. Ведуться два проекти, орієнтовані на підтримку просторів індивідуальних даних. Перший з них – проект SEMEX (SEMantic Explorer) [5, 6] – виконується в University of Washington під керівництвом Хелеві. Другий, з назвою iMeMex [7], виконується під керівництвом Йенса-Петера Диттриха в ETH Zurich. Проте, судячи з аналізу інтернет-джерел, жоден з проектів ще не формалізував поняття простору даних, що, у свою чергу, призводить до розрізненості підходів роботи з ними.

Важливим елементом інтеграції є сумісне використання структурованих, частково структурованих та неструктурованих джерел інформації. Як показано у [7], наразі проблема пошуку неструктурованої інформації вирішується лише в окремих областях, для яких побудована онтологія.

Отже, *метою* статті є формалізація поняття простору даних та визначення операцій над ним, а також розроблення методів інтеграції неоднорідної інформації. Для цього розглянемо основні елементи простору даних, формалізуємо методи взаємодії між ними та розробимо методи автоматичного визначення структур даних джерела.

ФОРМАЛІЗАЦІЯ ПРОСТОРУ ДАНИХ

1. Подання простору даних як алгебраїчної системи

Як відомо [8], *алгебраїчною системою* $\langle \boldsymbol{Al}; \boldsymbol{WF}; \boldsymbol{WR} \rangle$ називається об'єкт, що складається з трьох множин: непорожньої множини ***Al***, множини операцій алгебри ***WF***, визначених на ***Al***, і множини відношень (предикатів) ***WR***, визначених на ***Al***:

$$\boldsymbol{A} = \langle \boldsymbol{Al}, \boldsymbol{WF}, \boldsymbol{WR} \rangle. \quad (1)$$

Дослідження в області моделей даних інформаційних систем [3, 4] показують, що на сьогодні центральним стало поняття типу даних. З цим зв'язані як проблематика створення нових мов програмування, так і впровадження сучасних технологій організації даних, зокрема і просторів даних.

Будь-який інформаційний простір E доцільно подати у вигляді абстрактної алгебраїчної системи (1), де ***Al*** – об'єкти інформаційного простору; ***WR*** – зв'язки між об'єктами ***Al***; ***WF*** – операції маніпулювання об'єктами у просторі. Як об'єкти моделі (1) можуть виступати компоненти інтелектуальної системи – файли всіх типів, каталоги, логічні і фізичні диски.

Відношення $\boldsymbol{WR} = \{WR1, \ldots, WR_n1\}$ між об'єктами інформаційного простору визначає конкретну конфігурацію інтелектуальної системи, орієнтовану на конкретного користувача чи користувачів, $\boldsymbol{G} = \{G1, \ldots, G_n2\}$ – множина користувачів. Модель взаємодії користувача з інформаційним простором можна подати у вигляді:

$$Y(t) = E(Z1(t), \ldots, Z_n2(t)),$$

де $Z_i(t)$ – вхідний вплив на інформаційний простір з боку користувача $G_i \in \boldsymbol{G}$; $Y(t)$ – реакція системи (відповідь), що сконфігурована під користувача і має вигляд E. У загальному випадку $Z_i(t)$ – елементарна задача, що користувач G_n вирішує за допомогою інформаційного простору $E(\boldsymbol{Al}, \boldsymbol{WR}, \boldsymbol{WF})$. Прикладами елементарних задач є: пошук інформації (за зразком, за індексом, за описом, за методом найближчого сусіда тощо), інтеграція даних (консолідація, федералізація, розповсюдження), агрегація тощо [9].

У загальному випадку кожна із елементарних задач вирішується на певному носії даних Al_j, $j = 1, \ldots, n3$, з використанням певних операції маніпулювання WF_j, ефективність виконання яких для задачі $Z_i(t)$ залежить від типу носія. Користувач не знає наперед, з яким саме носієм йому потрібно працювати, та дозволені операції над цим носієм. Тому визначення типу елементарної задачі відбувається за допомогою множини відношень ***WR***.

Множина відношень ***WR*** здійснює структуризацію знань про носій інформаційного простору та допустимі операції над ним.

Визначимо правила структуризації даних довільної предметної області:

– факторизація множини об'єктів інформаційного простору ***Al*** за відношенням еквівалентності [3];

– конструювання додаткових функцій *Id, Num, Selector*:

Id(*x*) – функція задає для кожного об'єкту додатковий атрибут – його індивідуальний ідентифікатор;

Num(*x*) – функція задає для кожного об'єкту додатковий атрибут – його порядковий номер в класі еквівалентності X_i, де $i = 1, \ldots, p$. Областю значення функції *Num* є множина натуральних чисел;

Selector(*x*) – функція задає для кожного об'єкту додатковий атрибут – його подання. Областю значень для цієї функції є деякий кортеж з атрибутів об'єкту, тобто значень функцій *Id*(*x*), *Num*(*x*), $f_1(x)$, $f_2(x)$, …, $f_k(x)$;

– побудова інвертованих індексів [4];

– побудова багатовимірних матриць (використання алгебри кортежів).

Оскільки інструмент моделювання баз даних повинен з потреби включати не лише засоби структуризації даних, але і операційні можливості для маніпулювання даними, модель даних в інструментальному сенсі розуміється як алгебраїчна система.

Основними моделями для побудови інформаційних систем є бази даних, сховища даних, простори даних.

Подамо кожен із зазначених об'єктів як алгебраїчну систему.

2. Побудова ієрархії об'єктів носіїв простору даних

Отже, реляційна база даних – це алгебраїчна система, у якої носієм є множина реляційних відношень ***r***, множиною операцій – реляційна алгебра $\Re$, множиною предикатів – словник даних (схема даних бази даних) ***R***.

$$\boldsymbol{DB} = <\boldsymbol{r}, \Re, \boldsymbol{R}>, \Re = \{\pi, \sigma, \triangleright\!\triangleleft \cup, \cap, - \}. \qquad (2)$$

Тепер дамо формальне означення сховища даних.

Сховищем даних (СД) назвемо шістку

$$\mathbf{DW} = <\mathbf{DB}, \boldsymbol{rf}, \boldsymbol{RF}, \boldsymbol{rm}, \boldsymbol{RM}, \boldsymbol{func}>,$$

де **DB** – множина вхідних баз даних (реляційних, багатовимірних, об'єктно-орієнтованих, ненормалізованих тощо) (або множина відношень, їх схем та обмежень цілісності, які містять інформацію з вхідних баз даних), ***rf*** – множина відношень фактів, ***RF*** – схема ***rf***, ***rm*** – множина відношень метаданих, ***RM*** – схема ***rm***, ***func*** – множина процедур прийняття рішень.

Метадані – дані, що містять опис структури сховища даних, джерел та приймачів даних тощо (дані про дані). Тоді *нові дані* (або *рішення*) – це результат застосування функцій сховища даних над відношенням фактів:

$$Design = func(rf, user\_param),$$

де *user_param* – множина параметрів користувача, або вимог, які ставляться до рішення.

Відношення між вимірами – відношення, яке є зв'язком між певними вимірами та відношенням фактів:

$$V_1 \times V_2 \times \ldots \times V_n \times rf \to rel.$$

У відношенні фактів виміри подаються за допомогою зовнішніх ключів, а самі значення – за допомогою атрибутів агрегації. У свою чергу, *rel* можуть бути параметрами для інших відношень між вимірами і тим самим створювати ієрархію вимірів.

Над даними сховища даних виконуються такі операції:

1. *Інтеграція даних* – це об'єднання даних, які знаходяться у різних системах (базах даних). Існують такі методи інтеграції:

– консолідація даних – це збір даних з територіально віддалених або різноплатформенних джерел DB_i даних в єдине сховище даних DW з метою їх подальшого опрацювання та аналізу.

$$DW.rel \xrightarrow{consolid} DB_1.\boldsymbol{r} \cup \ldots \cup DB_n.\boldsymbol{r}.$$

– операція *федералізації даних* полягає у витяганні даних з первинних систем на підставі зовнішніх вимог. Всі необхідні перетворення даних здійснюються при їх витяганні з первинних файлів.

$$\text{Virtual.DW} : \sigma_{fed\ \boldsymbol{rm} = DB1.\boldsymbol{r}}(DB_1.\boldsymbol{r}) \cup \ldots \cup \cup \sigma_{fed\ \boldsymbol{rm} = DBn.\boldsymbol{r}}(DB_n.\boldsymbol{r}).$$

2. *Агрегація даних* – це обчислення узагальнених значень на основі даних відношень вимірів для підтримки стратегічного або тактичного керування з детальних даних.

$$\boldsymbol{rel} = Ag(DB1.\boldsymbol{r}, \ldots, DBn.\boldsymbol{r}).$$

Опишемо сховище даних як алгебраїчну систему.

Оскільки воно інтегрує інформацію з баз даних, а інтегровані значення містяться у відношенні фактів, то звідси випливає, що сховище даних – це алгебраїчна система виду

$$\mathbf{DW} = <\boldsymbol{DW}, \aleph, \boldsymbol{rm}>, \quad \boldsymbol{DW} = \{\boldsymbol{rel}, DB_1.\boldsymbol{r}, \ldots, DB_n.\boldsymbol{r}\},$$
$$\aleph = \{\Re, \xrightarrow{consolid}, \sigma_{fed}, Ag, func\}. \qquad (3)$$

Отже, алгебраїчна система класу реляційна БД є підсистемою алгебраїчної системи класу сховище даних.

Тепер дамо формальне означення простору даних.

Простір даних **DS** – це множина даних, поданих у різних моделях (баз даних **DB**, сховищ даних **DW**, статичних Web-сторінок **Wb**, неструктурованих даних **Nd**, графічних та мультимедійних даних **Gr**), ло-

кальних сховищ та **ODW**, а також засобів інтеграції **Int**, пошуку **Se** та опрацювання інформації **Wo**, об'єднаних середовищем керування моделями **EM** [10, 11].

$$\mathbf{DS} = <\mathbf{DB}, \mathbf{DW}, \mathbf{ODW}, \mathbf{Wb}, \mathbf{Nd}, \mathbf{Gr}, \mathbf{Int}, \mathbf{Se}, \mathbf{Wo}, \mathbf{EM}>. \quad (4)$$

Каталог **CG** – це реєстр ресурсів даних, що містить найбільш базову інформацію про кожного з них: джерело, ім'я, місцезнаходження в джерелі, розмір, дату створення і власника та ін. Каталог є інфраструктурою для більшості інших сервісів простору даних, але він також може підтримувати базовий, призначений для користувача, інтерфейс перегляду простору даних.

Для організації роботи з розрізненими джерелами використовуються словник термінів та понять (ключових слів) *Dic*, який містить синонімічний опис одного і того ж концепту у різних джерелах даних. Заповнення словника даних на початку здійснюється за допомогою розробленої онтології предметної області, пізніше – автоматизовано.

$$\text{Metadata}(\mathbf{DB}, \mathbf{DW}, \mathbf{Wb}, \mathbf{Nd}, \mathbf{Gr}, \mathbf{ODW}) \cup \cup Dic \Rightarrow \mathbf{Cg}. \quad (5)$$

Для подання простору даних як алгебраїчної системи необхідною умовою є уніфікація джерел даних, оскільки саме вони є носіями (об'єктами, над якими виконуються операції та відношення алгебраїчної системи). Уніфікація сховищ даних та баз даних здійснюється за допомогою інтелектуального агента (подано нижче). Проте, як видно із визначення простору даних (4), джерелами його інформації є також неструктурований текст та веб-сайти. Для ефективного пошуку та аналізу неструктурованої текстової інформації використаємо семантичну мережу.

Семантична мережа – це структура для подання знань у вигляді вузлів, з'єднаних дугами. Особливості структури семантичних мереж:

1) вузли семантичних мереж являють собою концепти предметів, подій, станів, які у свою чергу визначаються із словника *Dic*;

2) довільні вузли одного концепту відносяться до різних значень, якщо вони не відмічені як такі, що відносяться до одного концепту;

3) дуги семантичних мереж створюють відношення між вузлами-концептами (помітки над дугами вказуватимуть на тип відношення).

Визначимо семантичну мережу неструктурованого джерела інформації Q як двійку

$$Q = \{V, D\},$$

де $V = \{v_i\}$ – множина вершин (вузлів мережі), $V \in Dic$, $D = \{d_j\}$ – множина дуг.

Дуги між елементами визначають взаємозв'язки між вершинами і задають послідовність пошуку концептів (їх важливість). Вершини є елементами локального сховища даних **ODW.**

Для опису веб-ресурсів використовують поняття семантичної павутини, функції та структура якої співмірні з семантичною мережею. Для створення зрозумілого комп'ютеру опису ресурсу в семантичній павутині використовується формат RDF. Оскільки джерелами даних простору даних є веб-ресурси, то для *Dic* використовуватимемо формат RDF. Пошук у такій мережі здійснюватиметься за допомогою ключових слів.

Побудуємо функцію трансформації неструктурованого тексту та веб-сайтів у вигляді семантичної мережі:

SemNet(**Wb**) → **ODW –** для веб-ресурсів,

SemNet(**Nd**) → **ODW –** для текстових даних.

Подання неструктурованих даних у вигляді семантичної мережі із збереженням вершин та відношень між ними у локальному сховищі **ODW** дозволяє звести інформацію з неоднорідних джерел даних до баз даних та сховищ даних, що, у свою чергу, при визначенні та уніфікації їхніх структур даних дасть можливість здійснювати інтеграцію, пошук та агрегування даних.

3. Агент визначення структури джерела

Визначення структур даних джерел просторів даних здійснюється за допомогою інтелектуального агента

$$\text{EM}(\mathbf{CG}) \xrightarrow{Agent} \mathbf{ODW}. \quad (6)$$

Агент *Op* подається сімкою об'єктів [11]:

$$Agent = = <\mathbf{CG}, \text{EM}, Dic, Experience\_Base, Solver, Effector>, \quad (7)$$

де **CG** – ідентифікатор внутрішнього стану агента (інформація про джерела, що вже є у ПД); EM – компонента агента, що відповідає за сприйняття середовища (сенсор), тобто середовище керування моделями; *Dic* – база знань, що містить знання агента про власні можливості (терміни-синоніми, що позначають у джерелах одні і ті ж властивості); *Experience_Base* – база накопиченого досвіду агента, що містить «історію» впливів на агент з боку середовища й відповідної їм реакції агента ($Experience\_Base = \sigma_{evdate = Date()}(Dic)$); *Solver* – компонента, що відповідає за навчання (подає список розбіжностей,

які виявив агент); *Effector* – компонента, яка відповідає за дії агента (формування запиту по декількох джерелах, приведення результатів запитів по джерелах до єдиної структури, відмова у запиті).

В основі роботи агента лежить інформація про джерела, які вже є у просторі. Його задачею є порівняння структур даних джерела даних, що входитиме у простір, з структурами даних джерел, що вже є у просторі, та визначення різниці. Це дозволить автоматизувати формування запитів, що виконуватимуться у просторі даних. Чим більше джерел здатний «розрізнити» агент, тим точніше буде інформація в **ODW** і тим ефективніше можна буде проводити процедури інтеграції, пошуку та опрацювання даних у просторі даних **DS**.

Розглянемо принцип роботи агента порівняння інформації із двох схем даних для тих самих фізичних сутностей. При цьому допускається, що схеми мають різні системи кодування, тобто той самий об'єкт може мати в цих схемах різні ідентифікатори. Допускається, що назви таблиць, атрибутів і розподіл атрибутів у таблицях можуть розрізнятися. Але передбачається, що між схемами існують взаємозв'язки, які можуть бути задані експертами (словник *Dic*). Необхідно класифікувати типи можливих взаємозв'язків і знайти необхідні умови для інтеграції даних на основі цих взаємозв'язків.

Нехай деяка сутність описується в першій схемі даних відношенням A, що містить кортежі $\{x_1, x_2, \ldots, x_n\}$, а в другій схемі даних відношенням B, що містить кортежі $\{y_1, y_2, \ldots, y_m\}$. Відношення A і B можуть бути як окремими таблицями в реляційній схемі даних, так і переглядами. Запишемо формально умову, що A і B містять ті самі фізичні сутності. Будемо вважати, що в цьому випадку існують взаємозв'язки між окремими атрибутами x_i й y_j. Розглянемо різні типи таких взаємозв'язків між двома скалярними атрибутами x і y, визначеними на скінчених доменах X і Y відповідно.

1. Змістовний взаємозв'язок доменів. Найзагальнішим типом взаємозв'язку можна вважати випадок, коли ми хоча б можемо визначити, чи співпадають об'єкти за атрибутами x і y або не співпадають і чи співпадають назви-синоніми у словнику термінів *Dic*. Інакше кажучи, задана функція змістовної еквівалентності: $P: X \times Y \to \{0, 1\}, Dic_{X=Y}$. $P(x, y) = 1$, якщо за атрибутами x і y об'єкти співпадають, $P(x, y) = 0$ у іншому випадку. Якщо $P(x, y) = 1$ і $Dic_{X \neq Y}$, то доповнюємо *Dic* новими синонімами.

2. Існує відображення, що конвертує X в Y, якщо для будь-якого $x \in X$ значення існує $y \in Y$ значення, таке, що за атрибутами x і y об'єкти будуть співпадати. Інакше кажучи, існує відображення, $F: X \to Y$, таке, що для всіх $x \in X$ виконується рівність

$$P(x, F(x)) = 1, \; Dic_{X \neq Y}. \qquad (8)$$

3. Існує узагальнююче відображення з X в Y (Y – узагальнення X), якщо для будь-якого значення $x \in X$ існує рівно одне значення $y \in Y$, таке, що за атрибутами x і y об'єкти будуть співпадати. Інакше кажучи, існує відображення $F: X \to Y$, таке, що для всіх $x \in X$ виконуються умова (2.5) і нерівність

$$P(x, y) < 1, Dic_X, Dic_Y \text{ для всіх } y \neq F(x). \qquad (9)$$

4. Існує узагальнююче відображення X на Y (X – деталізація Y), якщо для будь-якого значення $x \in X$ існує рівно одне значення $y \in Y$, і для будь-якого Y існує хоча б одне значення x, таке, що за атрибутами x і y об'єкти будуть співпадати. Інакше кажучи, існує відображення $F: X \to Y$, таке, що для всіх $y \in Y$ існує $x \in X$, такий, що $F(x) = y$; і для всіх $x \in X$ виконуються умови (8) і (9).

Крім наведених типів взаємозв'язків, розглянемо наступні:

а) існує відображення, що конвертує Y в X.

б) існує узагальнююче відображення з Y в X.

в) існує узагальнююче відображення Y на X.

Будемо вважати, що об'єкт, заданий кортежем $a = \{x_1, x_2, \ldots, x_n\}$ в одній схемі даних, співпадає з об'єктом, заданим кортежем $b = \{y_1, y_2, \ldots, y_m\}$ в іншій схемі даних, якщо вони співпадають за всіма взаємозалежними атрибутами, тобто для всіх функцій взаємозв'язку відношень $P_{ij}: X_i \times Y_j \to \{0, 1\}$ правильна рівність $P_{ij}(x_i, y_j) = 1$. Множину пар індексів (i, j), для яких задані функції P_{ij}, позначимо $\Omega = \{(i, j)\}$, $i = Num(x)$, $j = Num(y)$, $x, y \in Dic$. Тоді можна задати функцію відповідності об'єктів $P: A \times B \to \{0, 1\}$ таким чином:

$$P(a, b) = 1, \text{ якщо } P_{ij}(x_i, y_j) = 1 \text{ для всіх } (i, j) \in \Omega; \qquad (10)$$

$$P(a, b) = 0, \text{ якщо існує } (i, j) \in \Omega, \text{ такі, що } P_{ij}(x_i, y_j) \neq 1. \qquad (11)$$

Перейдемо до класифікації взаємозв'язків між схемами даних.

1. Відповідність об'єктів. Якщо Ω не порожня, і задана функція $P: A \times B \to \{0, 1\}$, будемо говорити, що встановлено відповідність об'єктів. Нехай X_1 і Y_1 є первинними ключами відношень A і B. Тоді, якщо вибрати всі пари $\{x_1, y_1\}$, для яких $P(a, b) = P(\{x_1, x_2, \ldots, x_n\}, \{y_1, y_2, \ldots, y_m\}) = 1$, одержимо таблицю відповідності *Dic* із заголовком $\{\langle x_1 : X_1 \rangle, \langle y_1 : Y_1 \rangle\}$.

Маючи таку таблицю, можна робити запити, що одержують дані з обох схем, таким чином:

Select $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m$

From A, B, *Dic*

Where $Dic.X_1 = A.X_1$ **and** $Dic.Y_1 = B.Y_1$

2. За кортежем a з відношення A можна швидко знайти у відношенні B кортеж b такий, що $P(a, b) = 1$, не створюючи й не використовуючи таблицю відповідності.

3. За кортежем з A можна однозначно визначити кортеж в B.

4. Відношення A і B синхронізовані. Якщо за кортежем з A можна однозначно визначити кортеж в B і за кортежем з B можна однозначно визначити кортеж в A, будемо говорити, що відношення A і B синхронізовані. Зміст цієї умови в тому, що якщо перенести деякий кортеж з A в B, а потім назад, то гарантовано не буде створено нового запису, що дублює a.

Отже, результатом роботи агента є встановлення взаємозв'язку між схемами даних.

4. Операції над носіями простору даних

Одною з ключових задач побудови простору даних є визначення виразної потужності запитів із **Se**. Над носіями простору даних виконуються такі операції із множини **Se**:

1) *Запит про довільні дані* $\mathrm{Se}_{\mathrm{simple}}$ – у користувачів повинна бути можливість запиту будь-якого елементу даних, незалежно від його формату і моделі даних. Здійснюється на основі ключових слів key_word та каталогу даних **CG**.

$$\mathrm{Se}_{\mathrm{simple}} : \sigma_{key\_word(Cg)}. \qquad (12)$$

2) *Стуктуровані запити* будуються з використанням SQL та подібних мов. За допомогою каталогу визначається, чи джерело, у якому здійснюватиметься пошук, містить структуровану інформацію. Якщо це так, то виконується запит безпосередньо до джерела даних. У іншому випадку запит продовжується виконуватись по каталогу даних у вигляді пошуку ключових слів.

$$\mathrm{Se}_{\mathrm{structured}} : \sigma_{key\_word(Cg)}, \sigma(\mathrm{Source}). \qquad (13)$$

3) *Запити до метаданих* повинні забезпечуватися можливості:

– отримання даних про джерело відповіді та місцезнаходження джерела;

– визначення елементів даних в просторі даних, що можуть залежати від заданого елементу даних, і підтримка гіпотетичних запитів;

– визначення рівня невірогідності відповіді.

$$\mathrm{Se}_{\mathrm{meta}} : \sigma_{user\_param}(Cg), \qquad (14)$$

де *user_param* – множина параметрів користувача (вимог до запиту), його профілю, або вимог, які ставляться до рішення.

Простір даних є не тільки засобом обміну даними. Він повинен містити засоби отримання нових знань. У контексті просторів даних знання – це результат застосування засобів опрацювання даних над джерелами та каталогом даних:

Design = **Wo**(**DB**, **DW**, **Wb**, **Nd**, **Cg**, *user_ param*).

Під профілем користувача будемо розуміти підмножину каталогу даних, яка вказує на ті джерела даних, до яких користувач має доступ.

$$\mathrm{profile} : \sigma_{\mathrm{access} = \mathrm{Yes}}(\mathbf{Cg}).$$

Із визначення простору даних випливає подання ПД як алгебраїчної системи:

$$\mathbf{DS} = \langle \boldsymbol{DS}, \wp, \mathbf{Cg} \rangle,$$

$$\boldsymbol{DS} = \{\mathbf{ODW}.\boldsymbol{r}, \mathrm{DW1}.\boldsymbol{rel}, \ldots, \mathrm{DWn}.\boldsymbol{rel}, \mathrm{SemNet}(\mathbf{Wb}), \mathrm{SemNet}(\mathbf{Nd})\},$$

$$\wp = \{Agent(\aleph), Se_{simple}, Se_{structured}, Se_{meta}, \sigma_{access}, Agent\}. \qquad (15)$$

Таке визначення ґрунтується на таких висновках:

– базу даних можна вважати виродженим сховищем даних (сховище даних з єдиним джерелом та обмеженою множиною операцій – реляційною алгеброю),

– оскільки інформація про інші джерела простору даних (**Wb**, **Nd**, **Gr**) міститься у каталозі **Cg** (побудова семантичної мережі), а дані, що отримуються з цих об'єктів, за допомогою операцій інтеграції потрапляють у локальне сховище даних **ODW**, то в просторі даних **Wb**, **Nd**, **Cr** можна замінити каталогом даних **Cg**.

Отже, алгебраїчна система класу сховище даних та алгебраїчна система класу реляційна база даних є підсистемами алгебраїчної системи класу простір даних.

5. Операції над просторами даних

Простори даних можуть вкладатися одне в інше (наприклад, простір даних району вкладається в простір даних області), і вони можуть перекриватися (наприклад, простір даних в сфері туризму перекривається з просторами даних оздоровчо-лікувальної, історичної сфери та сфери управління природними ресурсами).

Тому в просторі даних повинні міститися правила розмежування доступу. Прикладами таких розмежувань для простору даних в сфері туризму є:

– для учасників простору даних в сфері туризму надати можливість пошуку даних у просторах даних оздоровчо-лікувальної, історичної сфери та сфери управління природними ресурсами;

– для учасників простору даних сфери управління природними ресурсами надати права блокування записів та встановлення властивості неактуальності для даних простору даних в сфері туризму та ін.

Уведемо операцію об'єднання просторів даних:

$$DS_1 \cup DS_2 = <\mathbf{DB}_1 \cup \mathbf{DB}_2, \mathbf{DW}_1 \cup \mathbf{DW}_2, \mathbf{Wb}_1 \cup \mathbf{Wb}_2, \mathbf{Nd}_1 \cup \mathbf{Nd}_2, \mathbf{Mp}_1 \cup \mathbf{Mp}_2, \mathbf{ODW}_1 \cup \mathbf{ODW}_2, \mathbf{Int}, \mathbf{Se}, \mathbf{Wo}_1, \mathbf{Wo}_2, \mathbf{EM}>,$$

$$\mathbf{Cg} = \text{profile}(\mathit{Agent}(\mathbf{Cg}_1) \cup \mathit{Agent}(\mathbf{Cg}_2)),$$

$$\mathbf{Int} = \mathbf{Int}_1 = \mathbf{Int}_2,$$

$$\mathbf{Se} = \mathbf{Se}_1 = \mathbf{Se}_2,$$

$$\mathbf{EM} = \mathbf{EM}_1 = \mathbf{EM}_2.$$

Уведемо операцію перетину просторів даних:

$$DS_1 \cap DS_2 = <\mathbf{DB}_1 \cap \mathbf{DB}_2, \mathbf{DW}_1 \cap \mathbf{DW}_2, \mathbf{Wb}_1 \cap \mathbf{Wb}_2, \mathbf{Nd}_1 \cap \mathbf{Nd}_2, \mathbf{Mp}_1 \cap \mathbf{Mp}_2, \mathbf{ODW}_1 \cap \mathbf{ODW}_2, \mathbf{Int}, \mathbf{Se}, \mathbf{Wo}, \mathbf{EM}>,$$

$$\mathbf{Cg} = \mathbf{Cg}_1 \cap \mathbf{Cg}_2,$$

$$\mathbf{Wo} = \mathbf{Wo}_1 \cap \mathbf{Wo}_2,$$

$$\mathbf{Int} = \mathbf{Int}_1 \cap \mathbf{Int}_2,$$

$$\mathbf{Se} = \mathbf{Se}_1 \cap \mathbf{Se}_2,$$

$$\mathbf{EM} = \mathbf{EM}_1 = \mathbf{EM}_2.$$

ВИСНОВКИ

У статті подано формальну модель простору даних. Показано, що алгебраїчні системи класу база даних та сховище даних є підкласом алгебраїчної системи класу простір даних.

Наукова новизна полягає у поданні простору даних як алгебраїчної системи. Уведено операції над просторами даних.

Практична цінність полягає у визначенні основних задач та компонент простору даних та зв'язку між ними.

Подальші дослідження стосуватимуться формалізації методів пошуку неструктурованих, напівструктурованих та строго структурованих даних та побудови відповідних алгоритмів.

СПИСОК ЛІТЕРАТУРИ

1. Интеграция данных и хранилища [Електронний ресурс] : за даними InterSoft Lab. – 2006. – Режим доступу: http://citcity.ru/12101/
2. Интеграция корпоративной информации: новое направление [Електронний ресурс] : за даними InterSoft Lab. – 2006. – Режим доступу:http://citcity.ru/11155/
3. *Qi Su.* Indexing Relational Database Content Offline for Efficient Keyword-Based Search / Qi Su, Jennifer Widom // 9th International Database Engineering; Application Symposium (IDEAS'05). – 2005. – P. 297–306.
4. *Аграновский А. В.* Индексация массивов документов / Аграновский А. В., Арутюнян Р. Э. [Електронний ресурс]. – 2003. – Режим доступу: http://www.scandocs.ru/page.jsp? pk=node_1185787748359.
5. *Denoyer L.* The Wikipedia XML Corpus / Denoyer L., Gallinari P. // SIGIR Forum. – 2006. – P. 108–121.
6. DeRose P. DBLife: Acommunity information management platform for the database research community / DeRose P., Shen W., Chen F., Lee Y., Burdick D., Doa A., Ramakrishnan R. // In CIDR. – 2007. – P. 92–101.
7. *Dong X.* A Platform for Personal Information Management and Integration / Dong X., Halevy A. / In CIDR. – 2005. – P. 67–71.
8. *Мальцев А. И.* Алгебраические системы / Мальцев А. И. – М. : Наука, 1970. – 392 стр.
9. *Шаховська Н. Б.* Простори даних: поняття та призначення // Матеріали конференції CSIT-2007. – Львів, 2007. – С. 269–277.
10. *Шаховська Н. Б.* Особливості моделювання просторів даних // Комп'ютерна інженерія та інформаційні технології : вісник НУ «Львівська політехніка». – 2008. – № 608. – С. 145–154.
11. *Шаховська Н. Б.* Простір даних області наукових досліджень // Моделювання та інформаційні технології. – 2008. – № 45. – С. 132–140.

Надійшла 29.04.2009
Після доробки 04.06.2009

Шаховская Н. Б.

ФОРМАЛИЗАЦИЯ ПРОСТРАНСТВА ДАННЫХ С ПОМОЩЬЮ АЛГЕБРАИЧЕСКОЙ СИСТЕМЫ

Проанализированы проблемы обработки данных из различных источников. Описана формальная модель пространства данных и операции над ним.

Ключевые слова: пространство данных, хранилище данных, база данных, алгебраическая система, поиск данных, группировка данных, интеллектуальный агент, источник данных.

Shakhovska N. B.

FORMALIZATION OF DATA SPACE USING THE ALGEBRAIC SYSTEM

Problems of different sources data processing are analyzed. A formal data space model and operations performed in it are described.

Key words: data space, data warehouse, database, algebraic system, database search, classification, intellectual agent, data source.