

Процедура „вставляння повернення продажів” використовує процедуру „оброблення продажу для повернення”, алгоритм якої представлений на рис. 10.

Алгоритми оброблення процедури „вставляння повернення коштів” і процедури „оброблення оплати для повернення” представлені на рис. 11–12.

Алгоритм оброблення процедури „оброблення залишків” представлений на рис.13.

У процесі оброблення залишків забезпечуються можливості погашення заборгованості за параметрами, які вказуються при внесенні оплат:

- погашення боргу у межах дат реалізації товарів (початок і кінець погашення боргу);
- якщо для Покупця встановлена можливість погашення боргу поза межами початку і кінця погашення боргу, забезпечується можливість погашення таких боргів;
- заборгованість поштучних товарів може погашатись деякою кількістю оплат.

Процес оброблення проходить у два етапи:

– на першому етапі переглядаються залишки кредитів, для кожного кредиту переглядаються всі залишки неоплачених товарів і погашаються борги за визначеними параметрами при внесенні оплати;

– на другому етапі переглядаються залишки неоплачених товарів, для кожного залишку визначається загальна сума кредитів Покупця і погашається борг, якщо виконуються умови погашення боргу.

### **Висновки**

Отже, використовуючи той факт, що Visual FoxPro – це об’єктно-орієнтована мова, яка керується за подіями, можна створити базу даних розрахунків, що працює в режимі реального часу і забезпечує повну інформацію про стан розрахунків Покупця з Продавцем

**УДК 51.001.57+004.652.4+004.827**

**В.В. Пасічник, Н.Б. Шаховська**  
Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## **ІНФОРМАЦІЙНІ КОМПОНЕНТИ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ**

© Пасічник В.В., Шаховська Н.Б., 2008

**Показано відмінності між базами, сховищами та просторами даних. Введено означення інженерії знань та інтелектуальної системи.**

**Differences are shown between databases, datawarehouses and dataspaces. D of engineering of knowledges and intellectual system is entered.**

### **Вступ**

Інтелектуальні системи є одними із найпоширеніших засобів підтримки прийняття рішень. Проте сьогодні немає чіткого визначення цього терміна, опису складових таких систем та їхніх завдань. Тому подамо ряд визначень, які характеризують інтелектуальні системи, та охарактеризуємо їх складові.

У зв'язку з поширеним різночитанням та трактуванням поняття інформаційної технології (ІТ) подамо сутнісну структуру цього терміну, використовуючи при цьому термінологічні статті популярного інформаційного ресурсу, яким є Wikipedia - (<http://www.wikipedia.org/>).

**Технологія** (від грецького *téchne* – мистецтво, майстерність, вміння та грецького *logos* – знання), сукупність методів та інструментів для досягнення бажаного результату, спосіб перетворення чогось заданого в необхідне. Технологія – це наукова дисципліна, у межах якої розробляються та удосконалюються способи й інструменти виробництва.

У широкому розумінні – це знання, які можна використати для виробництва продуктів (товарів та послуг) з економічних ресурсів. У вузькому розумінні технологія подається як спосіб перетворення речовини, енергії, інформації в процесі виготовлення продукції, обробки та переробки матеріалів, складання готових виробів, контроль якості та керування.

Технологія містить методи, прийоми, режими роботи, послідовність операцій та процедур, вона тісно взаємопов'язана із засобами, що застосовуються, обладнанням, інструментами, використовуваними матеріалами. За методологією ООН – технологія в чистому вигляді охоплює методи та техніку виробництва товарів і послуг (*dissembled technology*, англ.); втілена технологія охоплює машини, обладнання, споруди, виробничі системи та продукцію з високими техніко-економічними параметрами (*embodied technology*, англ.). Матеріальна технологія (МТ) створює матеріальний продукт. Інформаційна технологія (ІТ) створює інформаційний продукт на основі інформаційних ресурсів.

**Інформаційні технології (ІТ)** використовують комп'ютерні та програмні засоби для реалізації процесів відбору, реєстрації, подання, збереження, опрацювання, захисту та передавання інформації – інформаційного ресурсу у формі даних та знань – з метою створення інформаційних продуктів.

**Інформаційну систему (ІС)** подаємо як певний набір інформаційних технологій, що в комплексі зорієнтовані на досягнення певної системної мети, виконуючи задані функції та пропонуючи при цьому споживачам якісні інформаційні продукти та сервіси.

Своєю чергою, для всіх штучних інформаційних систем притаманними є чотири життєвих фази їхнього формування та функціонування. Йдеться про фази системного аналізу, системного проектування, системної інтеграції та системного адміністрування, які генерують відповідні вимоги до професійної підготовки та практичної орієнтації фахівців у царині інформаційних систем. Ринок потребує системних аналітиків, системних проєктувальників, системних інтеграторів та системних адміністраторів.

Як відомо, для роботи інформаційної системи необхідною умовою є наявність інформації.

У загальному випадку існує три типи існування фактів, ідей, уявлень про певні об'єкти, суб'єкти, події проєктованої предметної області:

– інформація – об'єктивізовані відомості, які існують незалежно від нас та від наших уявлень про них;

– дані – інформація, подана у формалізованому (матеріалізованому) вигляді, придатному для застосування в комп'ютерних інформаційних системах;

– знання – суб'єктивізовані відомості, отримані від працівників-експертів певної предметної області, і які подають певні залежності, послідовності, висновки.

Своєю чергою, дані подаються у формі:

– структури даних;

– баз даних;

– сховищ даних;

– просторів даних.

– Знання подаються у вигляді:

– баз знань;

– дедуктивних баз даних;

– експертних систем тощо.

Форми подання даних утворюються як пари структур даних та функцій їх опрацювання.

Дані сприймаються їх отримувачем як певна змістовна інформація лише в тому випадку, коли в його «пам'яті» закладені поняття і моделі, що допомагають зрозуміти зміст отриманих відомостей.

Коли дані опрацьовуються певними методами або евристичними, то на їх основі можна встановити залежності, послідовності чи висновки, які дають змогу підтримати процес прийняття рішення. Такі дані називають знаннями. Рішення може прийматися як експертом предметної області (такі знання називають суб'єктивними), так і комп'ютером (такі знання називають комп'ютеризованими).

Зв'язок між інформацією, даними та знаннями подано на рис. 1.

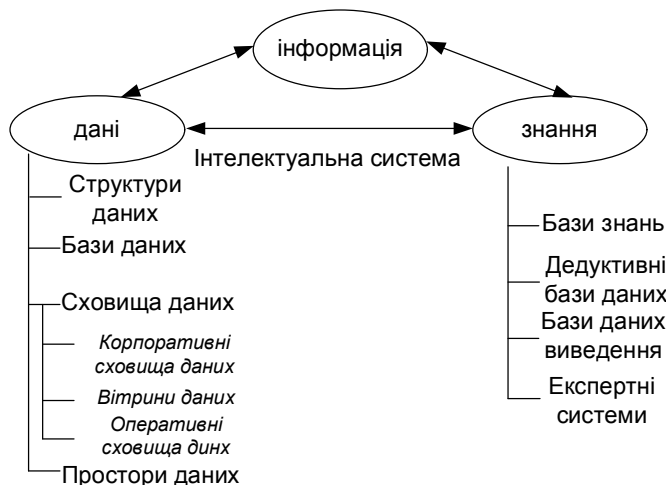


Рис. 1. Зв'язок між інформацією, даними та знаннями

Стаття присвячена визначенню зазначених понять, вказанню відмінностей між ними та встановленню їх місця в інтелектуальних системах.

### Основний матеріал

Визначимо форми подання даних.

Наведемо декілька найпоширеніших визначень бази даних (БД).

**Означення 1. Базу даних** можна визначити як сукупність взаємопов'язаних даних (прості чи складені типи), що зберігаються разом на одному носії та описують якусь предметну область за наявності такої мінімальної надмірності, яка допускає їх оптимальне використання для одного або декількох застосувань. Розрізняють ієрархічні, мережеві, реляційні, часові (темпоральні), постреляційні (об'єктно-орієнтовані, з гніздуванням), розподілені та багатовимірні бази даних.

Реляційною базою даних називають трійку:

$$DB = \langle r, R, Z \rangle,$$

де  $r$  – множина відношень бази даних,  $R$  – множина їх схем,  $Z$  – множина обмежень цілісності.

Використання бази даних припускає роботу з нею декількох прикладних програм (застосувань), що вирішують завдання різних користувачів.

**Означення 2. Сховище даних** – це агрегований інформаційний ресурс, що містить консолідовану інформацію з усієї проблемної області та використовується для підтримки прийняття рішень.

**Означення 3. Консолідована інформація** – це одержані з декількох джерел та системно інтегровані різноманітні інформаційні ресурси, які в сукупності наділені ознаками повноти, цілісності, несуперечності та становлять адекватну інформаційну модель проблемної області з метою її аналізу, опрацювання та ефективного використання в процесах підтримки прийняття рішень.

Виходячи з означення, подамо формальну модель сховища даних (СД). Ним назвемо шістьку

$$DW = \langle DB, rf, RF, rm, RM, func \rangle,$$

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.

Типове сховище даних зазвичай відрізняється від традиційної реляційної бази даних. По-перше, традиційні бази даних призначені для того, щоб допомогти користувачам виконувати повсякденну роботу, тоді як сховища даних призначені для прийняття рішень. Наприклад, продаж товару і виписування рахунку здійснюються з використанням бази даних, призначеної для опрацювання транзакцій, а аналіз динаміки продажів за декілька років, що дає змогу спланувати роботу з постачальниками, — за допомогою сховища даних.

По-друге, традиційні бази даних характеризуються постійними змінами у процесі роботи користувачів, а сховище даних відносно стабільне: дані у ньому зазвичай оновлюються за розкладом (наприклад, щотижня, щодня або щогодини — залежно від потреб). В ідеалі процес поповнення (або як надалі його називатимемо – завантаження) є просто додаванням нових даних за певний період часу без зміни попередньої інформації, що вже міститься у сховищі.

І, по-третє, традиційні бази даних найчастіше є джерелом даних, що потрапляють до сховища. Крім того, сховище може поповнюватися за рахунок зовнішніх джерел, наприклад статистичних звітів. Дані, що надходять до бази даних з іншої бази, є невеликого обсягу (тисячі записів), мають ту саму схему даних, що і база даних-приймач. На відміну від них сховища даних у визначені терміни отримують значно більші обсяги даних, які можуть відрізнятися від приймача форматом, а інколи і типом, що вимагає застосування додаткових процедур трансформування та завантаження даних (так звані процедури ETL – Extract, Transform, Load).

Як бази даних, так і сховища даних можуть будуватись на основі певної системи керування базами даних (СКБД) (реляційна, постреляційна тощо). СКБД забезпечує загальний репозиторій для зберігання і опрацювання структурованих даних. СКБД підтримує набір взаємозв'язаних послуг і дозволяє розробникам зосередитись на специфічних проблемах їх застосувань, а не на завданнях, які виникають за потреби в узгодженому й ефективному керуванні великими обсягами даних. Проте СКБД вимагають, щоб всі дані знаходилися під єдиним адміністративним керуванням і відповідали єдиній схемі. У відповідь на задоволення цих обмежень СКБД можуть забезпечити розвинені засоби маніпулювання даними та опрацювання запитів зі зрозумілою і строгою семантикою, а також строгі транзакційні гарантії оновлень, паралельного доступу і довготривалого зберігання (так звані властивості ACID).

Враховуючи специфіку, сховище даних має такі особливості проектування та побудови:

– отримання інформації з різних джерел даних (зокрема і з реляційних баз даних) у деталізованому та агрегованому вигляді (зберігаються результати застосування функцій агрегації – суми, середнього значення, максимуму, мінімуму тощо);

– багатовимірне подання інформації – ігноруються деякі вимоги нормалізації (дотримують максимум 3-ї нормальної форми), що значно підвищує швидкість опрацювання інформації, оскільки зменшує кількість операцій з'єднання JOIN;

– наявність метаданих для опису джерел метаданих та структури самого сховища даних – у базах даних також використовують словники для опису структур даних, а у сховищах даних метадані (словники, дані про дані) повинні будуватися за класифікаційною схемою Захмана. За цією схемою описують об'єкти (що?), суб'єкти (хто?), місцезнаходження (де?), час (коли?), фактори впливу, чинники (чому?), способи (як?);

– наявність пакетного завантаження даних в сховище даних та вивантаження даних;

– наявність процедур аналізу даних та отримання нових даних;

– орієнтованість даних на аналітичне, а не на статичне опрацювання.

Сховища даних краще пристосовані до зберігання та аналітичного опрацювання великих обсягів даних і в основному є інтеграцією реляційної та багатовимірної моделей. Сьогодні є такі архітектури побудови сховищ даних: корпоративна інформаційна фабрика Білла Інмона, шина Ральфа Кімбола, зведення даних корпорації TDAN. Вони мають розвинені засоби інтеграції даних з різних джерел та дають змогу працювати як з деталізованою, так і агрегованою інформацією.

Парадигма для реляційних даних у сховищі даних (**парадигма корпоративної інформаційної фабрики КІФ – Corporate Information Factory, CIF**) розроблена Інмоном і передбачає, що дані повинні перебувати на низькому ступені деталізації і в третій нормальній формі (ЗНФ, 3NF). Білл Інмон підтримує повторний або спіральний підхід до розвитку великого сховища даних. За цим підходом розвиток сховища відбувається ітераційно, тобто у разі виникнення потреби додається одна таблиця за один раз, що забезпечує лише незначну зміну схеми даних. Тому такий підхід до проектування сховища ще називають спіральним підходом.

Є альтернативний підхід до архітектури сховищ даних, відомий як **сховище даних з архітектурою шини, або підхід Ральфа Кімбола, або Просторове Сховище**. У цій моделі первинні дані перетворюються в інформацію, придатну для використання, на етапі підготовки даних. При цьому обов'язково беруться до уваги вимоги до швидкості опрацювання інформації та якості даних. Як і в моделі Білла Інмона, підготовка даних починається зі скоординованого добування даних із джерел. Ряд операцій відбувається централізовано, наприклад, підтримка і зберігання загальних довідкових даних, інші дії можуть бути розподіленими.

Область подання просторово структурована, при цьому вона може бути централізованою або розподіленою. Просторова модель сховища даних містить ту саму атомарну інформацію, що й нормалізована модель, але інформація структурована по-іншому, щоб полегшити її використання й виконання запитів. Ця модель містить як атомарні дані, так і узагальнювальну інформацію (агрегати у зв'язаних таблицях або багатомірних кубах) відповідно до вимог продуктивності або просторового розподілу даних. Запити в процесі виконання звертаються до усе нижчого рівня деталізації без додаткового перепрограмування з боку користувачів або розроблювачів застосування.

На відміну від підходу Білла Інмона, просторові моделі будуються для обслуговування бізнес-процесів (які, своєю чергою, пов'язані з бізнес-показниками або бізнес-подіями), а не бізнес-відділів. Наприклад, дані про замовлення, які повинні бути доступні для загальнокорпоративного використання, вносяться до просторового сховища даних тільки один раз, на відміну від КІФ-підходу, за яким їх довелося б тричі копіювати у вітрини даних відділів маркетингу, продажів і фінансів. Після того, як у сховищі з'являється інформація про основні бізнес-процеси, консолідовані просторові моделі можуть видавати їхні перехресні характеристики. Матриця корпоративного сховища даних з архітектурою шини виявляє й підсилює зв'язок між показниками бізнес-процесів (фактами) і описовими атрибутами (вимірами).

Гібридна архітектура, яка поєднує особливості реляційної та багатовимірної моделей, запропонована Дугласом Хекні. Іншою назвою моделі є **Узгоджувана вітрина даних**.

За такої архітектури передбачається подвійне проектування схеми сховища даних – розроблення нормалізованого центрального (корпоративного сховища) та багатовимірних (побудованих за архітектурою шини) вітрин даних. Корпоративне нормалізоване сховище дає змогу коректно зберігати дані, а ненормалізовані вітрини – швидко виконувати запити користувачів

**Зведення даних (Data Vault) (ЗД)** – предметно орієнтована, історична й унікально зв'язана множина нормалізованих таблиць, які підтримують одну або більше функціональних предметних областей. Це – гібридний підхід, що поєднує кращі особливості 3-ї нормальної форми (ЗНФ) і схеми «зірка». Модель гнучка, масштабується, послідовна і пристосована до потреб різних предметних областей. Вона відповідає потребам сховища даних і відкидає потребу у використанні вітрин даних та, на відміну від гібридного підходу Хекні, не вимагає подвійної роботи для надбудови архітектури шини над архітектурою корпоративної фабрики.

Зведення даних може керувати масивними наборами гранульованих даних в меншому, більш нормалізованому фізичному просторі, наприклад ЗНФ і схемі «зірка». Ґрунтується на математичних

принципах, які підтримують нормалізовані моделі даних. Внутрішня частина моделі зведення даних – близькі структури, які відповідають традиційним визначення схеми «зірка» і ЗНФ, що містять виміри, зв'язки “багато до багатьох” і стандартні табличні структури. Відмінності полягають у поданні зв'язків, структуризації поля і гранульованому, пов'язаному з часом, зберіганні даних.

Є такі підвиди сховища даних: вітрина даних, оперативне сховище даних.

**Означення 4. Вітрина даних (ВД)** — зріз сховища даних, масив тематичної, вузьконапрямленої інформації, що орієнтований, наприклад, на користувачів однієї робочої групи або департаменту.

Дворівнева архітектура сховища даних передбачає побудову вітрин даних без створення центрального сховища, при цьому інформація надходить із реєстраційних систем і обмежена конкретною предметною областю. При побудові вітрин використовуються основні принципи побудови сховищ даних, тому їх можна вважати сховищами даних у мініатюрі.

Побудова повноцінного корпоративного сховища даних зазвичай виконується в тривірневій архітектурі. На першому рівні розташовані різноманітні джерела даних – внутрішні реєструвальні системи, довідкові системи, зовнішні джерела (дані інформаційних агентств, макроекономічні показники). Другий рівень містить центральне сховище, куди стікається інформація від всіх джерел з першого рівня і, можливо, оперативне сховище даних (сховище поточної інформації, розглянуто далі), що не містить історичних даних і виконує дві основні функції:

- по-перше, воно є джерелом аналітичної інформації для оперативного керування;
- по-друге, тут підготовляються дані для подальшого завантаження в центральне сховище.

**Означення 5. Операційне сховище даних (ОСД)** – це предметно-орієнтований, інтегрований, змінюваний набір консолідованих даних, який містить поточну (не історичну) деталізовану інформацію.

На перший погляд, операційне сховище даних дуже схоже на сховище за структурою і змістом. Зазвичай за деякими характеристиками ОСД і сховище даних дуже схожі, але ОСД має ряд властивостей, які істотно відрізняють його від сховища. Як ОСД, так і сховище даних є предметно-орієнтованим інтегрованим набором консолідованих даних. З цього погляду вони схожі, оскільки як в одному, так і в іншому випадку дані повинні бути завантажені з транзакційних систем. Але на цьому їх схожість закінчується. ОСД містить дані, що змінюються, тоді як в сховищі дані після завантаження не змінюються. Інша відмінність полягає у тому, що операційне сховище містить тільки дані, актуальні на певний момент часу, тоді як в сховищі містяться як поточні, так і історичні дані. При цьому актуальність даних у сховищі значно нижча, ніж в операційному сховищі. Як правило, у сховищі містяться дані, завантажені протягом останніх 24 годин, тоді як актуальність даних в ОСД може вимірюватися секундами. Ще однією відмінністю ОСД від сховища є те, що в ньому містяться тільки детальні дані, тоді як сховище містить як детальні, так і агреговані дані.

І бази даних, і сховища даних дають змогу опрацювати деталізовані та інтегровані дані, що побудовані на основі наперед визначених моделей даних. У випадку роботи у всесвітній мережі з величезною кількістю ресурсів (прикладом таких задач є туристичний бізнес – збирання інформації про місця відпочинку, її інтеграція та зберігання у внутрішніх базах даних, геоінформаційні системи – сьогодні ще не розроблено єдних стандартів подання такої інформації, а її збирання також відбувається із джерел з наперед не відомими моделями даних) неможливо визначити, які саме моделі даних використовуватимуться. Тому винятково за допомогою баз даних та сховищ даних не можна організувати ефективної взаємодії між усіма об'єктами у цих предметних областях. Розробники часто зустрічаються з набором слабо зв'язаних джерел даних і тому повинні кожного разу вирішувати низькорівневі завдання управління даними. До цих завдань входять забезпечення можливостей пошуку і запиту даних; дотримання правил, обмежень цілісності, угод про іменування тощо; відстежування походження даних; забезпечення доступності, відновлення і контролю доступу; керований розвиток даних і метаданих.

Традиційні СКБД представляють тільки одну точку (хоч і дуже важливу) в просторі рішень управління даними. Важливою точкою є "системи інтеграції даних". Насправді, системи інтеграції даних і обміну даними традиційно призначаються для підтримки багатьох інших служб у системах просторів даних. Особливість полягає у тому, що в системах інтеграції даних потрібна семантична інтеграція до того, як можуть бути забезпечені які-небудь інші послуги. Тому, хоч і відсутня єдина схема, якій відповідають всі дані, система повинна знати точні взаємозв'язки між елементами, що використовуються в кожній схемі. У результаті для створення системи інтеграції даних потрібна значна попередня робота.

Простір даних розглядають як нову абстракцію управління даними [1]. Як ключова задача робіт у області управління даними використовується платформа підтримки просторів даних (DataSpace Support Platforms, DSSP). DSSP забезпечує набір взаємозв'язаних послуг і гарантує розробникам можливість концентруватися на специфічних проблемах їх додатків, а не на завданнях, що повторюються, виникають за потреби узгодженої і ефективної роботи з взаємопов'язаними, але роздільно керованими даними.

На відміну від СУБД, в ядрі DSSP потрібна підтримка декількох моделей даних, щоб природно підтримувати якомога більше типів учасників.

**Означення 6. 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**), локальних сховищ та індексів **ODW**, а також засобів інтеграції **Int**, пошуку **Se** та опрацювання інформації **Wo**, об'єднаних середовищем керування моделями **EM**.

$$DS = \langle DB, DW, ODW, Wb, Nd, Gr, Int, Se, Wo, EM \rangle.$$

Як ключова задача робіт в області керування даними використовується платформа підтримки просторів даних (DataSpace Support Platforms, DSSP). DSSP забезпечує набір взаємозв'язаних послуг і гарантує розробникам можливість концентруватися на специфічних проблемах їх додатків, а не на завданнях, що повторюються; виникають за потреби узгодженої й ефективної роботи із взаємопов'язаними, але роздільно керованими даними.

На відміну від СКБД, в ядрі DSSP необхідна підтримка декількох моделей даних, щоб природно підтримувати якомога більше типів учасників.

Основні об'єкти простору даних та задачі, що розв'язуються з його допомогою, подано на рис. 2.

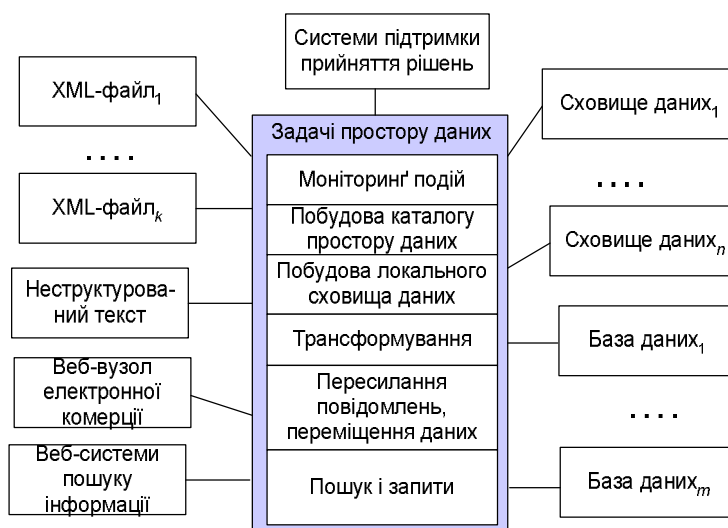


Рис. 2. Об'єкти простору даних та його задачі

Перші статті з опрацювання різних джерел даних та використання цих даних в єдиній предметній області з'явилися у 2005 р. У роботах [2, 3] задекларовано проблеми, які привели до необхідності введення такої абстракції даних, як простір даних. Серед них:

1. Інтеграція тексту, даних, коду і потоків (частково вирішена завдяки введенню Коддом 12-ти правил побудов сховищ даних та їх подальшої модифікації [12]).

2. Можливість багатоконтрольності даних (у концепції сховищ даних) забезпечувалась за допомогою процедур витягання, перетворення та завантаження даних (extract, transform and load – ETL) [3], а у випадку Інтернет із тисячами джерел інформації, поданої у різних форматах, ETL не придатна для використання).

3. Створення простих способів аналізу, узагальнення, пошуку і огляду електронних підбірок мультимедійної інформації, зокрема розроблення стандартів опису метаданих (сьогодні немає єдиних стандартів опису та опрацювання мультимедійної інформації).

4. Підтримка неточних та невчасних (тих, що надходять із запізненням або в неочікуваному порядку) даних та реалізація неточних запитів. Останніми роками достатньо інтенсивно досліджується окремий випадок цієї проблеми, так звані top-K-запити та неточні запити. Знову ж таки, розроблені моделі та технології працюють лише із визначеними моделями даних (зокрема реляційною – Fquery [12], Fuzzy Grouping та Fuzzy Lookup [13]). Стосовно невчасних даних взагалі відсутні методи, які б давали змогу не тільки фіксувати факт запізнення даних, але й на основі цих тимчасово відсутніх даних приймати рішення (ведуться роботи в області машин для обробки подій, проте сьогодні задекларовані лише проблеми, які призводять до задачі маніпулювання потоками. Роботи, що були проведені у середині 90-х років в області активних баз даних (Active DBMS, ADBMS), залишилися невикористаними через великий час відклику запитів та неврахування тимчасової відсутності даних [14]).

5. Релевантність відповіді повинна залежати від користувача і від контексту. Потрібне середовище для накопичення і використання відповідних метаданих. Є розробки щодо визначення типу користувача на основі записів у журналі доступу до ресурсів [14].

6. Проблема інтеграції даних, зокрема надоперативних (частково вирішена оперативними сховищами даних – дані збираються та оперативно опрацьовуються, але залежать від зовнішньої структури) та частково структурованих (частково вирішено засобами пошуку неструктурованих даних [10]).

7. Використання природних мов запитів до баз даних (так звані системи з природномовним інтерфейсом) [9] – передбачають формування запитів до системи у вигляді запитальних речень природною мовою.

8. Підтримка систем обробки поточкових даних (наприклад, система Postgres Майкла Стоунбрейкера, високорівнева мова “STREAMSQL” з вбудованими орієнтованими на потоки примітивами і операціями).

9. Повинна бути можливість ефективного зберігання, доступу і модифікації інформації про стан, а також її комбінування з реальними поточковими даними.

10. Для інтеграції в системі повинна використовуватися однорідна мова для роботи з усіма різновидами даних.

Як бачимо, майже за усіма вказаними напрямками ведуться роботи. Але ці роботи неінтегровані, не передбачають єдиного опрацювання та жорстко прив'язані до моделі даних, що є цілком неприйнятним у контексті просторів даних. Тому проблема формалізації просторів даних є актуальною.

Однією з основних служб простору даних є каталогізація елементів даних учасників.

**Означення 7. Каталог CG** – це реєстр ресурсів даних, що містить найбільш базову інформацію про кожний з них: джерело, ім'я, місцезоташування в джерелі, розмір, дата створення і власник і т.д. Каталог є інфраструктурою для більшості інших сервісів простору даних, але він також може підтримувати базовий, призначений для користувача, інтерфейс проглядання простору даних. Він не тільки містить описову інформацію (тобто виконує роль метаданих), але й зберігає для кожного учасника схему джерела, статистичні дані, швидкість зміни, точність, можливості відповідей на запити, інформацію про власника і дані, про політику доступу і підтримку конфіденційності. Оскільки джерела простору даних фізично не переносять у нього інформацію та



можуть обмінюватись між собою інформацією, то у каталозі необхідно зберігати дані і про зв'язки між джерелами.

Поверх каталога розміщене середовище управління моделями, яке дає змогу створювати нові зв'язки і маніпулювати існуючими зв'язками (наприклад, об'єднувати або інвертувати відображення, зливати схеми і створювати єдині представлення декількох джерел).

Важливою компонентою простору даних є компонента зберігання і індексування (ODW) для досягнення таких цілей:

§ для створення асоціацій між об'єктами даних від різних учасників;

§ для вдосконалення доступу до джерел з обмеженими власними засобами доступу;

§ для забезпечення можливості виконання деяких запитів без доступу до реального джерела даних;

§ для підтримки високого рівня доступності і відновлення.

Засоби індексування повинні володіти високим рівнем адаптивності до неоднорідних середовищ. Результатом локального зберігання та індексування є запит, що може повернути, наприклад, рядок в текстовому файлі, елемент шляху до файлу, значення в базі даних, елемент схеми або тег у XML-файлі. Важливими аспектами індексу є те, що, по-перше, він визначає інформацію для всіх учасників, коли деякі значення входять до декількох джерел даних (у деякому розумінні це узагальнює ідею індексів з'єднання). По-друге, індекс повинен справлятися з різноманітністю посилань на об'єкти предметної області, наприклад, з різними способами опису адміністративної одиниці.

Чим більше моделей здатне «розрізнити» середовище управління, тим точнішою буде інфорація в ODW і тим ефективнішими будуть процедури інтеграції, пошуку та опрацювання даних у просторі даних DS.

Оскільки одним із ключових питань простору даних є питання інтеграції, то розглянемо стандарти інтеграції.

Інтеграція інформаційних систем на основі веб-служб Int пов'язана з використанням чотирьох ключових стандартів [4]:

– Розширена мова розмітки інформації — Extensible Markup Language (XML). Описує інформацію, що пересилається по Інтернету. Запит на одержання яких-небудь даних чи виконання певних дій іншим застосуванням вимагає наявності способів передачі параметрів і одержання назад певних результатів. При використанні веб-служб ця інформація описується за допомогою мови XML, що є міжнародним загальноприйнятим стандартом для опису довільних даних, якими, своєю чергою, можуть обмінюватися інформаційні системи.

– Простий протокол доступу до об'єкта — Simple Object Access Protocol (SOAP). Цей стандарт описує протокол виклику веб-служби (віддалений процес доступу до послуг/інформації деякої прикладної системи). У типовій ситуації взаємодії система однієї організації може викликати систему іншої організації, використовуючи протокол SOAP. Запит, що зазвичай містить ту чи іншу форму бізнес-документа, посилається ініціатором до запитуваної системи. Остання приймає запит, і вхідний документ, який міститься в запиті, обробляється. У результаті запитувана система генерує відповідь, що повертається ініціатору взаємодії. Ініціатор також інформується про статус (успіх або інше) запиту.

– Мова опису веб-служб — Web Services Description Language (WSDL). Це мова, яка ґрунтується на стандарті XML, що визначає спосіб доступу до веб-служб. Вона описує функціональні можливості веб-служб і групує операції взаємодії у певні інтерфейси, що задають способи виконання операцій і ті параметри, які повинні бути на вході і виході.

– Універсальний метод опису, виявлення та інтеграції — Universal Description, Discovery, and Integration (UDDI). Технологія UDDI надає засоби, за допомогою яких можна зробити так, щоб будь-які застосування чи послуги, описані в термінах веб-служб, можуть бути розпізнані іншими застосуваннями та/або організаціями. Тобто це стандарт створення реєстра, використовуючи який можна описати організації і послуги, які вони надають, у вигляді, доступному для динамічного виявлення і взаємодії.

Засоби опрацювання даних Wo повинні підтримувати:

§ Видобування даних (Data mining) – асоціативні правила, дерева рішень, генетичні алгоритми тощо;

§ Засоби аналізу даних (Online Analytical Processing – OLAP) – реляційний OLAP (Relational OLAP – ROLAP), багатовимірний OLAP (Multidimensional OLAP – MOLAP), гібридний OLAP (Hybrid OLAP – HOLAP), динамічний OLAP (Dynamic OLAP – DOLAP);

§ Засоби природномовного пошуку – побудова нечітких запитів, запитів у вигляді природних питань, запитів до мета даних;

§ Засоби підбору контенту на основі аналізу характеристик користувача;

§ Засоби миттєвого аналізу даних (наприклад, визначення причин підвищення тиску у котлах за значеннями давачів приладів та пропонування методів усунення неполадок).

Отже, можна виділити такі особливості просторів даних [7]:

§ Простори даних складаються з широкої різноманітності форматів та інтерфейсів і усі без винятку формати даних повинні підтримуватися;

§ Дані у просторі даних не знаходяться під повним контролем;

§ Передбачається інтеграція тексту, даних, коду і потоків;

§ Підтримка структурованих, текстових, просторових, темпоральних, мультимедійних, процедурних даних; тригерів; потоків і черг даних як рівноправних компонентів;

§ Простори даних повинні забезпечувати вбудовану підтримку неточних даних. Повинна існувати можливість задання неточних запитів, і процесор запитів повинен ставитися до цього як до додаткового джерела неповноти і неточності;

§ Відповіді на запити повинні залежати від профілю користувача. Відповідь на запит експерта повинна відрізнятися від відповіді на запит новачка. Релевантність відповіді теж повинна залежати від користувача і від контексту;

§ Система повинна знати точні взаємозв'язки між елементами, що використовуються у кожній схемі;

§ DSSP пропонує рівні обслуговування та методи отримання приблизних відповідей;

§ DSSP повинен запропонувати інструменти і шляхи створення щільнішої інтеграції даних в просторі у міру необхідності.

Можуть забезпечуватися різні рівні послуг з обробки запитів до DSSP, і в деяких випадках вони можуть повертати якнайкращі з можливих приблизні відповіді. Наприклад, якщо деякі джерела даних стають недоступними, DSSP може забезпечити найкращий з можливих результат на основі даних, доступних під час виконання запиту.

Перейдемо до визначення поняття знання.

**Означення 8. Знання** — в широкому сенсі сукупність понять, теоретичних побудов і уявлень про предметну область загалом та про її суб'єкти, процеси тощо.

Знання — це закономірності предметної області (принципи, зв'язки, закони), одержані в результаті практичної діяльності і професійного досвіду, що дають змогу фахівцям ставити і розв'язувати задачі в цій області.

У контексті сховищ та просторів даних знання – це результат застосування функцій прийняття рішень над відношенням фактів:

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

де *user\_param* – множина параметрів користувача, або вимог, які ставляться до рішення.

Класифікація знань:

1. За природою. Знання можуть бути

– декларативні;

– процедурні.

Декларативні знання містять лише подання про структуру якихось понять. Ці знання наближені до даних, фактів. Наприклад: вищий навчальний заклад є сукупністю факультетів, а кожен факультет, своєю чергою, є сукупністю кафедр.

Процедурні ж знання мають активну природу. Вони визначають подання про засоби і шляхи отримання нових знань, перевірки знань. Це алгоритми різного роду. Наприклад, метод мозкового штурму для пошуку нових ідей.

2. За місцезнаходженням. Виділяють особові (неявні, приховані) знання і формалізовані (явні) знання.

✚ Неявні знання – це знання людей.

✚ Формалізовані (явні) знання:

- знання в документах;
- знання на компакт-дисках;
- знання в персональних комп'ютерах;
- знання в Інтернеті.

База знань (БЗ) призначена для збереження довгострокових даних, що описують розглянуту область (а не поточних даних), і правил, що описують доцільні перетворення даних цієї області. Містить знання, що належать до конкретної прикладної області, зокрема окремі факти, правила, що описують відношення чи явища, а також методи, евристики і різні ідеї, що стосуються розв'язання задач у цій прикладній області. У контексті сховищ та просторів даних це відношення фактів та метаданих.

**Означення 9. База знань** (англ. Knowledge base, KB) — це особливого роду база даних, розроблена для керування знаннями (метаданими), тобто збиранням, зберіганням, пошуком і видаванням знань. Розділ штучного інтелекту, що вивчає бази знань і методи роботи зі знаннями, називається інженерією знань.

Найважливіший параметр БЗ — якість знань, що містяться в ній. Кращі БЗ включають найбільш релевантну і свіжу інформацію, мають досконалі системи пошуку інформації та ретельно продуману структуру і формат знань.

Залежно від рівня складності систем, в яких застосовуються бази знань, розрізняють:

- ✚ БЗ всесвітнього масштабу — наприклад, Інтернет;
- ✚ БЗ національні — наприклад, Вікіпедія;
- ✚ БЗ галузеві — наприклад, Автомобільна енциклопедія;
- ✚ БЗ організацій — наприклад, поліклініки;
- ✚ БЗ експертних систем;
- ✚ БЗ фахівців.

Організація знань відбувається за допомогою формальної логічної моделі, продукційних правил, фреймів, семантичних мереж, генетичних алгоритмів.

Прості бази знань можуть використовуватися для зберігання даних про організацію: документації, керівництва, стану технічного забезпечення. Головна мета створення таких баз — допомогти менш досвідченим людям знайти існуючий опис способу вирішення якої-небудь проблеми предметної області.

База знань – важливий компонент інтелектуальної системи. Найвідоміший клас таких програм — експертні системи. Вони призначені для побудови способу вирішення спеціалізованих проблем, ґрунтуючись на записах БЗ і на призначеному для користувача описі ситуації.

**Означення 9.** Інженерія даних та знань (ІДЗ) – галузь інженерії з проблематикою інформаційного моделювання, моделей програмного забезпечення із забезпеченням адекватності, повноти та несуперечливості даних.

У літературі маємо такі визначення.

**Інженерія знань** – сукупність моделей, методів та технічних прийомів, метою яких є створення інтелектуальної системи.

ІДЗ – це область інформаційних технологій, мета якої – накопичувати і застосовувати знання, які не є об'єктом для опрацювання людиною, а об'єктом для опрацювання на комп'ютері. Для цього необхідно проаналізувати знання і особливості їх опрацювання людиною і комп'ютером, а також розробити їх машинне подання.

ІДЗ – дисципліна, пов'язана з питаннями:

- ✚ отримання знань від експертів і/або літератури, Internet;
- ✚ структуризації, формування і опрацювання знань;
- ✚ створення баз даних, баз знань, експертних систем, систем підтримки прийняття рішень.

Під отриманням знань розуміють процедуру взаємодії інженера за знаннями з джерелом знань, в результаті якої стають зрозумілими процес міркувань експертів при прийнятті професійних рішень і структура його уявлень про предметну область.

Термін «інженерія знань» запропонував Е. Фейгенбаум (E. Feigenbaum). За його визначенням інженерія знань – це теорія, методологія і технологія експертних систем, що охоплює методи видобування даних, аналізу знань та подання їх у вигляді правил. Головними елементами інженерії знань є дані та знання, подані у вигляді баз даних, баз знань, сховищ даних (реалізованих за допомогою СКБД) чи просторів даних (реалізованих за допомогою DSSP), опис яких подається за допомогою метаданих та метазнань.

**Означення 10. Інтелектуальна система (ІС)** – система, що ґрунтується на знаннях і використовується для підтримки прийняття рішень та отримання нових знань на базі методів інженерії знань.

Інтелектуальна система — інформаційна система, що використовує базу знань і розвинену систему програм її опрацювання.

Термін «інтелектуальна система» проілюстрований на рис. 3.

Класифікація завдань інтелектуальних систем подана нижче.

1.Інтерпретація даних. Це одне з традиційних завдань для експертних систем. Під інтерпретацією розуміється процес визначення сенсу даних, результати якого повинні бути узгодженими і коректними. Зазвичай передбачається багатоваріантний аналіз даних.

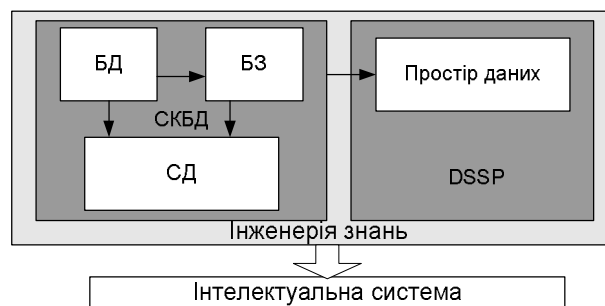


Рис. 3. Структура інтелектуальної системи

## Висновки

Стаття присвячена оглядові складових інтелектуальної системи та загальному аналізу галузі комп'ютерингу.

**Наукова новизна** статті полягає у встановленні формальних відмінностей між такими об'єктами, як бази даних, сховища даних та простори даних.

**Практичне значення** статті полягає у визначенні основних задач компонент інтелектуальної системи даних та зв'язку між ними.

1. Кузнецов С. *От баз данных к пространствам данных: новая абстракция управления информацией.* – 2006, [http://www.citforum.ru/database/articles/from\\_db\\_to\\_ds](http://www.citforum.ru/database/articles/from_db_to_ds). 2. Дрюэк К. (Katherine Drewek). "Хранилища данных: сходство и различия подходов Билла Инмона и Ральфа Кимболла", 2005, <http://www.b-eye-network.com/view/743>. 3. Dan Linstedt. *Data Vaulttm overview the next evolution in data modeling.* – 2005, <http://www.tdan.com/i021hy01.htm>. 4. *Огляд технологій інтеграції*

інформаційних систем, 2006, <http://www.microsoft.com/ukraine/government/analytics/integration/technologies/overview.msp> 5. Кузнецов С. Пространства данных: исследовательский полигон или путь к новому поколению систем управления данными? <http://synthesis.ipi.ac.ru/sigmod/seminar/s20060420>. 6 Donald Kossmann, Jens-Peter Dittrich. Personal Data Spaces. [http://www.inf.ethz.ch/news/focus/res\\_focus/feb\\_2006/index\\_DE](http://www.inf.ethz.ch/news/focus/res_focus/feb_2006/index_DE). 7. Garretts Summary of Principles of Dataspace Systems, [http://aravaipa.eas.asu.edu/wiki/index.php/Garretts\\_Summary\\_of\\_Principles\\_of\\_Dataspace\\_Systems#Overview](http://aravaipa.eas.asu.edu/wiki/index.php/Garretts_Summary_of_Principles_of_Dataspace_Systems#Overview) 8. ETH - Databases and Information Systems – iMeMex, [www.dbis.ethz.ch/research/current\\_projects/iMeMex](http://www.dbis.ethz.ch/research/current_projects/iMeMex) 9. Processing of natural language queries to a relational database. Samsonova M, Pisarev A, Blagov M, <http://www.cs.dartmouth.edu/~brd/Teaching/AI/Lectures/Summaries/natlang.html> 10. Основные концепции и подходы при создании контекстно-поисковых систем на основе реляционных баз данных. [http://www.citforum.ru/database/articles/search\\_sys.shtml](http://www.citforum.ru/database/articles/search_sys.shtml). 11. Особенности построения хранилищ данных. <http://citforum.uar.net/seminars/cis99/sch.shtml> 12. Kasprzyk J., Ziolkowski A. Database Queries with Fuzzy Linguistic Quantifiers // IEEE Transactions on Systems, Man, and Cybernetics. SMC-16, 1996. – P. 512-529. 13. Fuzzy Grouping в Microsoft SQL Server 2005 <http://msdn.microsoft.com/msdnmag/issues/05/09/SQLServer2005/default.aspx> 14. Пелецишин А.М. Методи та алгоритми моделювання Web-систем // Вісник Держ. ун-ту "Львівська політехніка". 2000. – №406. – С.199–211. 15. Черняк Л. Машины для обработки событий. – Открытые системы #09/2006, <http://www.osp.ru/os/2006/09/3776498/p1.html> 16. Гіпертекстові Технології, <http://moodle.ukma.kiev.ua/mod/resource/view.php?id=1120>

УДК 519.15:621.372

**В.В. Різник, М.Т. Соломко\***

Національний університет "Львівська політехніка",

\*Європейський університет (Рівненська філія),

Технологічно-природничий університет, м. Бидгощ (Польща)

## **СИНТЕЗ КОМБІНАТОРНИХ СИСТЕМ ЗА ДОПОМОГОЮ БАГАТОВИМІРНИХ В'ЯЗАНОК**

© Різник В.В., Соломко М.Т., 2008

**Розглянуто методи синтезу комбінаторних конфігурацій (ВІВ-схем) за допомогою багатовимірних числових конструкцій – ідеальних кільцевих в'язанок (ІКВ). Запропоновано алгоритми перетворення багатовимірних ІКВ в класичні комбінаторні конфігурації. Розкриваються нові можливості застосування ІКВ у сучасній комбінаториці.**

**Methods for synthesis of combinatorial configurations (ВІВ-designs) by means of multi-dimensional numerical constructions – so-called Ideal Ring Bundles (ІКВ)s has been considered. There are proposed algorithms of transition the multi-dimensional ІКВ into classic combinatorial configurations. The methods discovers new possibilities for apply of the ІКВ into modern combinatorial theory.**

### **Вступ**

Комбінаторні моделі та системи широко застосовуються в технічній кібернетиці, інформаційно-виміривальній та обчислювальній техніці, радіотехніці, зв'язку, електротехніці, машинобудуванні, а комбінаторні методи використовуються в теорії кодування, математичній логіці, програмуванні, теорії планування експерименту, технічній та статистичній фізиці, економіці, кристалографії, біології. Тому актуальними є дослідження властивостей існуючих і пошук нових