

ників ніколи не замінить необхідних професіоналів). Щорічна підготовка і перепідготовка персоналу організації і залучення викладачів відповідної кваліфікації — це один із шляхів вирішення кадрового питання;

В розробці УІС дуже ефективно себе показують студенти відповідних спеціалізацій, але вони повинні працювати під керівництвом талановитих викладачів тривалий час (не менше року) і отримувати відповідні пільги (кредити, зменшення оплати за навчання, направлення від організації на стажування і продовження навчання за обміном в інших ВЗО світу). Викладачам також необхідно враховувати таке навантаження як наукову і практичну роботу зі студентами.

Існуючі управляючі інформаційні системи потребують суттєвих змін у структурі і методах управління організації, а їхня ціна складає від десятків тисяч до мільйонів американських доларів. Це вельми обмежує коло використання таких систем і ускладнює їхню адаптацію.

Створення УІС ВЗО вимагає щорічного виділення великих коштів (до 10 % від бюджету всієї організації) і великих організаційних змін і займає багато часу (від 3 до 10 років), що необхідно враховувати при складанні бюджету і перспективних планів. Невиконання планів будь-яким підрозділом ВЗО веде до зриву або взагалі припинення створення УІС.

Експлуатація і утримання інформаційних систем вимагає постійного вкладення фінансів (від 5 % коштовності обладнання і програмного забезпечення системи).

БАГАТОВИМІРНІ БАЗИ ДАНИХ У СИСТЕМАХ З ЖОРСТКИМИ ОБМЕЖЕННЯМИ

І. Чичкань (кафедра інформатики)

У системах з жорсткими часовими обмеженнями одною з головних проблем є форма збереження даних з прив'язкою до часу. На перший погляд можна застосовувати дедуктивні СКБД. Цю можливість розглянемо по суті.

За визначенням дедуктивну БД можна розглядати як органічне поєднання двох складових частин:

- екстенційної, що містить власне факти;
- інтенційної, що містить правила для логічного висновку нових фактів на основі екстенційної частини і запиту користувача.

При такому загальному визначенні SQL-орієнтовану реляційну СКБД можна віднести до дедуктивних систем. Дійсно, що собою являють визначені в схемі реляційної БД уявлення, як не інтенційну частину БД.

Головною відмінністю справжньої дедуктивної СКБД від звичної реляційної є те, що і правила інтенційної частини БД, і запити користувачів можуть містити рекурсію. Саме можливість рекурсії робить реалізацію дедуктивної СКБД дуже складною і здебільшого ефективно нерозв'язною проблемою.

Не станемо тут докладніше розглядати конкретні проблеми дедуктивних систем, застосовувані обмеження і використовувані методи. Відзначимо лише, що звичайно мови запитів і визначення інтенційної частини БД постають логічними. Ось тому дедуктивні БД часто і називають логічними. Існує прямий зв'язок дедуктивних БД із базами знань, адже інтенційну частину БД можна розглядати як БЗ. Більш того, навіть важко провести грань між цими двома поняттями; принаймні, загальної думки з цього приводу не існує.

Для реалізації дедуктивної СКБД звичайно застосовується реляційна система. Отака система виступає в ролі сховища фактів і виконавця запитів, що надходять із рівня дедуктивної СКБД. Між іншим, таке використання реляційних СКБД підвищує актуальність задачі глобальної (або спільної) оптимізації запитів, тому що звичайно при здійсненні одного запиту рівня дедуктивної БД генерується декілька запитів рівня реляційної БД.

У випадку, коли набір правил дедуктивної БД стає великий і їх неможливо розмістити в оперативній пам'яті, виникає проблема керування їхнім сховом і доступом до нього у зовнішній пам'яті. Тут знову ж може бути застосована реляційна система, але вже не так ефективно.

Розробка систем з жорсткими часовими обмеженнями вимагає розробки більш складних структур даних і інших умов вибірки.

Для таких систем потрібна СКБД, яка могла б зберігати і забезпечувати роботу з даними, що прив'язані до часу. Звичайні БД зберігають миттєвий знімок моделі предметної області. Будь-яка зміна в момент часу t якогось об'єкта призводить до недоступності стану цього об'єкта в попередній момент часу. Зауважимо, що в цей же час у більшості відомих СКБД попередній стан об'єкта зберігається в часопису змін, але можливість доступу до цих даних для користувачів закрыта.

Звичайно, можна ввести у відношення явний часовий атрибут і підтримувати його значення на рівні додатків. У більшості випад-

ків так це і роблять. Недарма в стандарті SQL з'явилися спеціальні типи даних *DATE* і *TIME*. Але в такому підході закладено декілька вад:

- СКБД не знає семантики часового поля відношення і не може контролювати коректність його значень;
- з'являється додаткова збитковість схову — попередній стан об'єкта даних зберігається і в основній БД, і в часописі змін;
- мови запитів реляційних СКБД не пристосовані для роботи з часом.

Існує окремий напрямок досліджень і розробок в області темпоральних БД. У цій області досліджуються питання моделювання даних, мови запитів, організація даних у зовнішній пам'яті тощо. Основна теза темпоральних систем полягає в тому, що для будь-якого об'єкта даних, створеного в момент часу t_1 і знищеного в момент часу t_2 , у БД зберігаються, і є доступними для користувачів, усі його стани в часовому інтервалі $[t_1, t_2]$.

Дослідження і побудови прототипів темпоральних СКБД звичайно виконуються на основі деякої реляційної СКБД. Як і у випадку дедуктивних БД, темпоральна СКБД — це надбудова над реляційною системою. Звичайно, це не кращий спосіб реалізації з погляду ефективності, але він простий і дозволяє виконувати досить глибокі дослідження.

У доповіді викладаються питання побудови постреляційної системи з темпоральними можливостями. Характерні особливості системи, що пропонується, полягають у такому:

- система керування пам'яттю підтримує історичні дані;
- запити можуть містити часові характеристики об'єктів.

ДОСЛІДЖЕННЯ СИСТЕМИ ВЗАЄМОДІЮЧИХ ПРОЦЕСІВ ЗА ДОПОМОГОЮ ТЕОРІЇ АВТОМАТІВ

М. Глибовець (кафедра інформатики)

Для аналізу функціонування складної мережі найважливішими є два класичних поняття: визначеності (досягнення, живучості) і блокування (жоден процес не може працювати). Проміжний стан мережі можна трактувати як певний стан керування автомату. Перехід зміни стану мережі приводить до зміни стану керування автомату. Перевагою такого підходу є можливість використання єдиного підходу для моделювання програм, процесів, які вони описують і способи взаємодії цих процесів. Програми будемо розглядати на рівні блок-схем. Враховуючи паралельний характер роботи мережі, ми