

## ІНФОРМАТИКА

УДК 517.6

### ЗАСТОСУВАННЯ ОБ'ЄКТНОГО ПІДХОДУ ДО ТОПОЛОГІЇ РОЗМІЩЕННЯ ДАНИХ У МЕРЕЖІ

**В. Вовк**

*Львівський національний університет імені Івана Франка  
вул. Університетська, 1, м. Львів, 79000, e-mail: vovk@franko.lviv.ua*

Питання опрацювання розподіленої у мережі комп'ютерів інформації залишаються актуальними незважаючи на досягнутий прогрес у цій галузі. Проте у більшості випадків дискутують саме щодо методик реалізації такого опрацювання, приймаючи топологію розміщення даних наперед заданою. У роботі показано, що можливість керування таким розміщенням за дотримання певних правил розробки моделей даних та процесів у широкому класі задач дає значні переваги в ефективності та надійності функціонування масштабної системи.

*Ключові слова:* об'єктно-орієнтований підхід; бази даних; інформаційні системи.

#### 1. ВСТУП

Очевидна безальтернативність принципу відкритості розподілених інформаційних систем (ІС) є головним аргументом "глобалізації" сьгоднішніх ідеологій їхньої розробки. Провідні технології у цій галузі DCOM, CORBA, SOAP та інші надають підкреслено універсальний інструментарій для організації взаємодії довільних компонентів гетерогенної системи через набір інтерфейсів. Тобто, ніяких попередніх припущень щодо структури інформації та способів її опрацювання не роблять. Відповідно, і реалізація таких технологій є складною і недешевою. З іншого боку, відомо, що будь-яка універсальність методики максимально виправдана лише за наявності засобів її ефективного обмеження (з одночасним спрощенням реалізації) в разі появи додаткової інформації про специфіку вихідної задачі. На жаль, згадані вище супертехнології таких засобів практично не мають, і часто розробники залучають їхні "супер" можливості без гарантій об'єктивної потреби у цьому.

Ми відійдемо від практики всезагальності підходу до розробки ІС і зробимо це за рахунок додаткових припущень щодо структури її даних. Припущення будуть полягати навіть не в конкретизації вигляду цієї структури, а лише в єдиному підході до принципів її побудови в усіх окремих піделементах системи, які містять частини загальної інформації. Виконання такого обмеження, зрозуміло, можливе лише за умови розробки нової ІС. Крім того, її специфікою є умова відсутності жорстких вимог замовника щодо топології розміщення даних у мережі комп'ютерів. Утім, ці обмеження є невеликою платою за можливість будувати легкорозширювані, прості в експлуатації і недорогі в розробці ІС.

## 2. ОБ'ЄКТНІ МОДЕЛІ ДАНИХ ТА ПРОЦЕСІВ

Далі опиратимемось на ідеологію застосування об'єктного підходу до розробки моделей даних та процесів інформаційних систем, яка детально описана в [1,2]. Коротко нагадаємо її головні тези.

Структуру предметної області задачі подають у вигляді ієрархії класів (типів), кожному з яких відповідає окремий об'єкт (таблиця бази даних) моделі даних. Далі, без обмеження загальності, усі викладки виконуватимемо на спрощеному опорному прикладі моделі ІС вищого навчального закладу (ВНЗ).

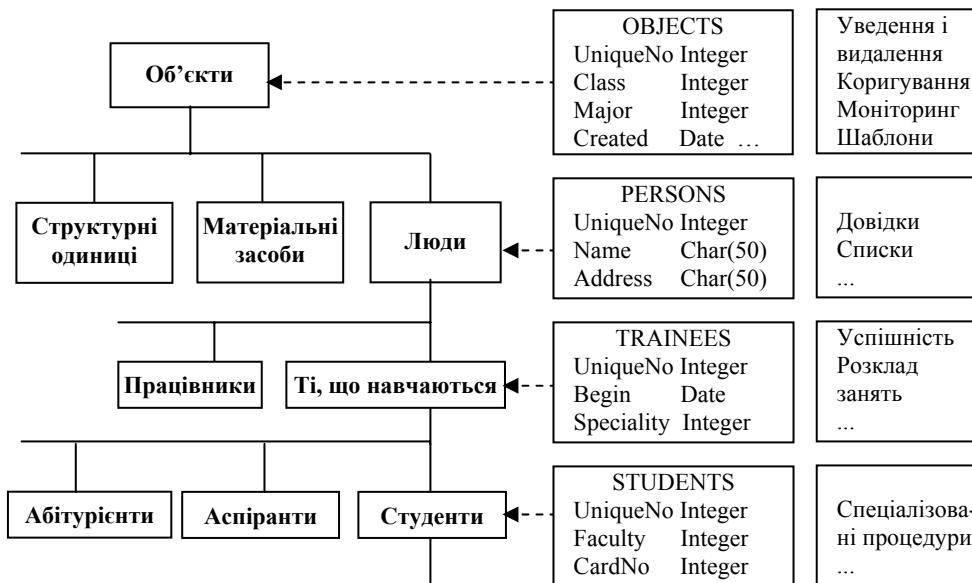


Рис.1. Фрагмент моделі даних і процедур ІС ВНЗ.

Процеси обробки даних поділяють на дві головні категорії: природні (за формулюванням задачі) та технологічні (подання інформації користувачам, тощо). На рис. 1 праворуч зображена гілка ієрархії природних процесів, яка є проекцією моделі даних, а реальні об'єкти, що наповнюють інформаційну систему, мають власну ієрархію відображення користувачу.

## 3. ДИНАМІЧНА ВЗАЄМОДІЯ КЛІЄНТІВ ТА СЕРВЕРІВ ЗАСТОСУВАНЬ

Типова реалізація багат шарової архітектури клієнт-сервер передбачає деревоподібну структуру, у якій кореневий шар становлять бази даних під керуванням відповідних систем керування базами даних (СКБД). Шар серверів застосувань виконує найбільшу обчислювано-затратну частину коду опрацювання інформації. Клієнти, здебільшого, лише відображають у зручному для користувачів вигляді отриману від серверів застосувань інформацію.

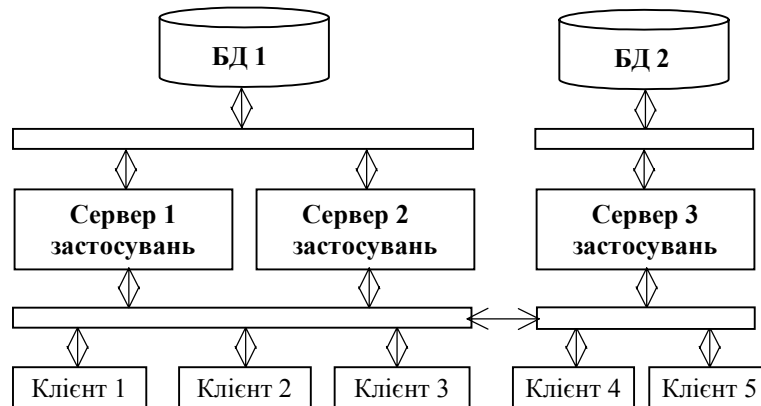


Рис. 2. Класична тришарова архітектура клієнт-сервер

Під номером сервера чи клієнта тут маємо на увазі його тип, а не конкретний екземпляр. Бази даних та відповідні сервери застосувань взаємодіють через локальну мережу. Клієнти входять у склад груп, що з'єднані між собою виділеними лініями або засобами Інтернету і, тому, здатні налагодити зв'язок з довільним сервером застосувань.

Програмний код опрацювання даних може міститись у самому сервері застосувань, або зберігатись у депозитаріях (як це, наприклад, передбачено технологією CORBA). Крім необхідності застосування недешевих технологій, за гнучкість другого підходу доводиться платити збільшенням завантаження мережі. Тому нижче розглянемо випадок серверів та клієнтів із вбудованими процедурами опрацювання даних.

Зробимо головне припущення, яке визначатиме подальшу стратегію побудови ієрархії виконавчих серверів:

**моделі даних і процесів усіх БД, що є в складі ІС, організовані за об'єктною методикою, що запропонована в [1,2].**

Наголосимо, що це припущення ніяк не обмежує ні змісту інформації і процедур її опрацювання в БД, ні типів керуючих СКБД, що важливо для функціонування гетерогенних ІС. Водночас воно є основою для створення єдиного для всіх клієнтських застосувань інтерфейсу навігації серед об'єктів системи, що означає значну уніфікацію процесів спілкування клієнта з серверами застосувань. Запропонована методика також задіює переваги об'єктного підходу у розподіленій розробці окремих елементів масштабної ІС.

Продемонструємо сказане на прикладі ІС ВНЗ. Нехай показана на рис. 3 БД1 є базою даних ректорату, БД2 + сервер застосувань розміщені у відділі кадрів, а БД3 з частиною серверів та клієнтів міститься у віддаленому корпусі, скажімо, фізичного факультету, який має зв'язок з локальною мережею головного корпусу через Інтернет. Згідно з прийнятим припущенням, об'єкти кожної БД становлять ієрархічну структуру, яка в інтерфейсі користувача [1] є деревоподібною. Тому клієнт, підєднуючись до будь-якого сервера застосувань (наперед налаштованого на визначену БД), відразу отримує дерево вузлів, як це, наприклад, зображено на рис. 4.

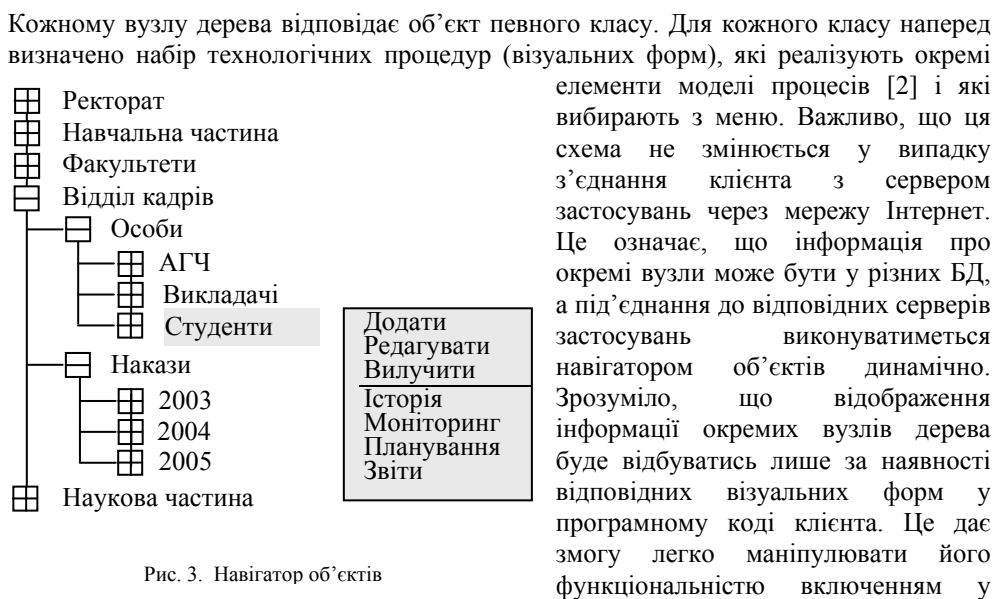


Рис. 3. Навігатор об'єктів

його склад лише потрібних елементів.

#### 4. ОБ'ЄКТНА ТОПОЛОГІЯ ДАНИХ ТА СЕРВЕРІВ ЗАСТОСУВАНЬ

Взаємодія серверів застосувань у типовій архітектурі клієнт-сервер (рис.2) не має системного характеру вже через те, що всіх їх вважають рівноправними і вони не утворюють ніякої ієрархії. Скористаємося припущенням про об'єктність моделей даних і процесів усіх складових частин нашої ІС. Для цього ще раз звернемося до рис. 1. Кожному класу об'єктів тут відповідає власна таблиця даних. Розділимо всі таблиці між окремими БД структурних одиниць установи, які уводять і коригують відповідну інформацію. Зрозуміло, що способи й ефективність такого поділу сильно залежать як від умов вихідної задачі, так і від досвіду розробника.

Нехай, наприклад, БД відділу кадрів містить інформацію в електронному вигляді ту ж, яку, зазвичай, зберігають у паперовому. Зате тепер ніде вона не дублюється, як це властиво паперовим технологіям (зазначимо, що повна інформація про особу далеко не обмежена відділом кадрів, а рознесена по БД інших підрозділів). Кожний підрозділ, що має БД, комплектується також власним сервером застосувань (або кількома), котрий спеціалізований на технологічне опрацювання відповідної частини даних (див. рис.4). Зрозуміло, що інформація класів найвищої ієрархії повинна зберігатись на загальноуніверситетських серверах і бути максимально доступною для клієнтів. Крім того, деякі підрозділи (наприклад, відділ аспірантури) навіть не маючи власної БД, можуть на власному сервері застосувань мати низку технологічних процедур опрацювання інформації.

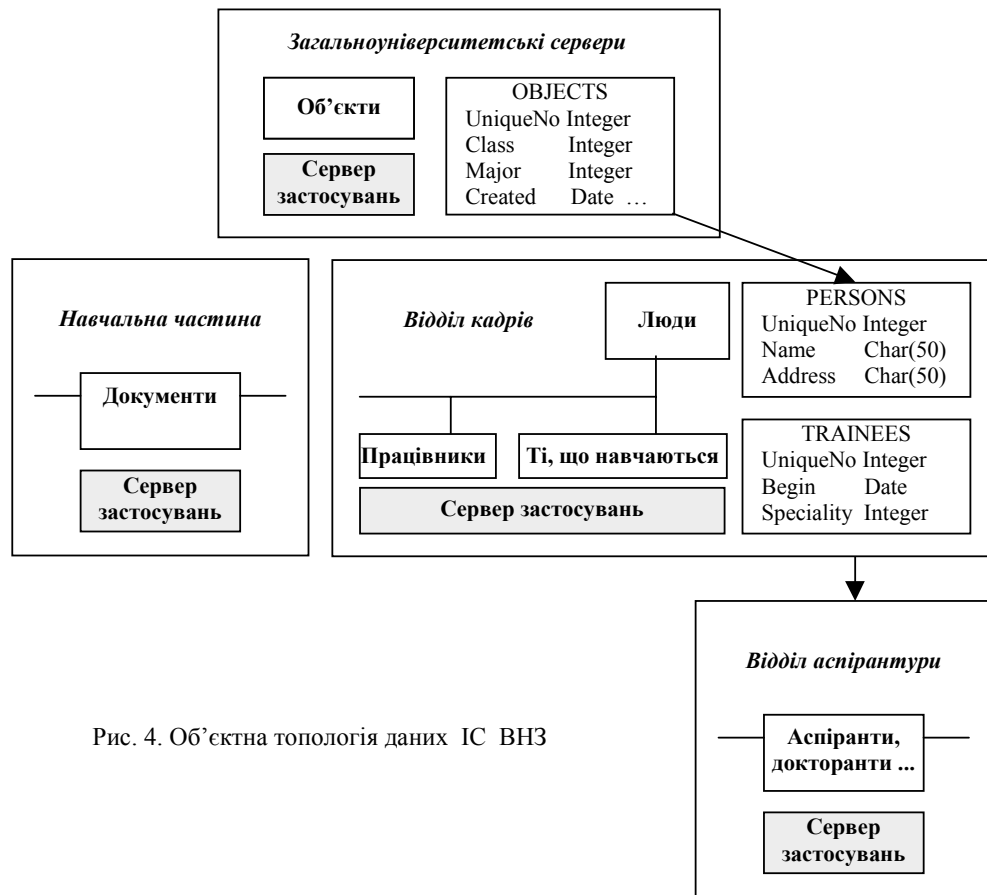


Рис. 4. Об'єктна топологія даних ІС ВНЗ

Наслідки запровадження цієї топології такі:

- позитивні
  - ІС у цілому є стійкішою до збоїв у роботі окремих підсистем, оскільки відсутність зв'язку з окремою БД (за винятком, можливо, OBJECTS) не порушує зв'язності решти інформації про об'єкт;
  - дані локалізовано у підрозділах, що дає змогу посилити відповідальність за їх збереження та супровід;
  - зменшуються обсяги дублювання інформації;
- негативні
  - для збирання інформації про об'єкт необхідне почергове з'єднання з декількома серверами застосувань, що може призводити до затримок у часі виконання складних запитів.

Окремим позитивним моментом, який зумисно виокремлений з наведеного перерахунку, є спадкова ієрархія, яку утворює програмний код серверів застосувань (унаслідок об'єктності моделей даних та процесів). Завдяки особливості тришарової

архітектури клієнт-сервер у перенесенні основної ваги програмного коду з клієнта на шар серверів застосувань цей факт обіцяє значні полегшення їхньої розробки та супроводу.

#### ЛІТЕРАТУРА

1. *Вовк В.Д.* Застосування об'єктного підходу до розробки інформаційних систем на основі реляційних баз даних // Вісн. Львів. ун-ту. Сер. прикл. математика та інформатика. 2000. Вип. 2. С. 184 – 190.
2. *Вовк В.Д., Мушкевич Б.О.* Застосування об'єктного підходу до розробки моделі процесів інформаційних систем з використанням реляційних баз даних // Вісн. Львів. ун-ту. Сер. прикл. математика та інформатика. 2001. Вип. 2. С. 64 – 69.
3. *Гради Буч.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++: 2-е изд./Пер. с англ. М.: Бином, СПб: Невский диалект, 1998. – 560 с.

#### AN APPLICATION OF THE OBJECT-ORIENTED APPROACH TO DATA TOPOLOGY DISTRIBUTION IN COMPUTERS NET

**V. Vovk**

*Ivan Franko National University in Lviv*  
*Universytetska str, 1, Lviv, 79000, e-mail: [kis@franko.lviv.ua](mailto:kis@franko.lviv.ua)*

Problems of processing of spatially shared information remain actual despite of essential progress in this area. However in most cases discuss techniques of realization of such processing, accepting topology of accommodation of the computers given in a network preset. In this paper is shown, that the opportunity of management of such accommodation at following in a wide class of problems gives the certain rules of development of models of the data and processes significant advantages in reliability of functioning of large system.

*Key words:* object-oriented approach; database; information systems.

*Стаття надійшла до редколегії 26.05.2005*

*Прийнята до друку 21.09.2005*