

ефективні вкладення паралельних програм в ОС з ієрархічною організацією комунікаційних середовищ. Актуальною є розробка не трудомістких методів і алгоритмів вкладення паралельних програм в ОС з ієрархічною організацією комунікаційних середовищ.

#### Література

1. Огневий О.В., Коваль Б.А., Присяжнюк В.В. Аналіз архітектурних особливостей обчислювальних систем з програмованою структурою / О.В. Огневий., В.А. Коваль, В.В.Присяжнюк // Зб. наук. праць Військового інст. Київського НУ ім. Тараса Шевченка. – К.: ВІКНУ, 2018. – Вип. № 48. – С. 143-148
2. Огневий О.В. Особливості управління інформаційними ресурсами в корпоративних інформаційних системах / О.В. Огневий., А.М.Огнева, В.А. Коваль, В.В.Присяжнюк // Вісник Хмельницького національного університету. - Хмельницький, 2018, № 6 - С.17-21

### **Модель опису структури об'єктно-орієнтованої бази даних для інформаційних систем**

Поліщук Ю.С., Стопчак О.О.

Науковий керівник: к.т.н. доц. Красильников С.Р.  
Хмельницький національний університет

Сучасний світ все більшою мірою набуває вигляд єдиного інформаційного простору. Велику роль в цьому процесі відіграє швидко розвиваючий інформаційний бізнес. Його бурхливий розвиток змушує розробників обчислювальної техніки шукати нові рішення, які дають можливість керувати складними структурами даних і надавати користувачеві доступ до них в кожному куточку земної кулі. Таким чином, інформація в сучасному світі перетворилася в один з найбільш важливих ресурсів. Не є винятком і інформаційні системи (ІС). Різноманітність завдань, що вирішуються за допомогою ІС, привело до появи безлічі різнотипних систем, але жодна з них немислима без використання баз даних.

Одним з основних компонентів ІС є інформаційне забезпечення, яке об'єднує різні складні структури даних. Такі структури зручно описувати у вигляді об'єктів, тому все більшої популярності в цій галузі стали займати об'єктно-орієнтовані бази даних (ООБД). ООБД дозволяють перенести у віртуальний світ об'єкти реального світу з мінімальними втратами. Однак, проектування ООБД є кропітким, трудомістким і динамічним процесом. Цей процес змістився з логічної області до абстрактного рівня, де важливим завданням стало конструювання коректної абстрактної моделі.

При цьому для побудови такої моделі традиційні методи проектування не підходять через наступних недоліків:

1) традиційні моделі даних не достатньо сильні в підтримці складних механізмів моделювання, таких як складові об'єкти і наслідування;

2) залежності обмежень, які використовуються в традиційному підході проектування (функціональні та багатозначні залежності) мають тільки обмежений набір залежностей, який може бути накладено на зв'язки і атрибути для вирішення питання пов'язаного з внутрішньо-об'єктними зв'язками. Крім того, залежність обмежень в плоскій і ущільненій моделях даних можуть бути класифіковані як обмеження, що характеризуються значенням. Сенс, який вони відображають, обумовлений взаємозв'язками між даними та відношеннями. На відміну від них, об'єктно-орієнтовані моделі даних підтримують ідею об'єктної ідентифікації, яка ідентифікує об'єкти незалежно від їх значень;

3) вони не підходять для розгляду проблем надмірності і проблем зміни на початку проектування об'єктно-орієнтованої схеми, тому що вирішують питання, пов'язане з отриманням результату, наприклад проблеми продуктивності та зберігання. Об'єктно-орієнтоване проектування даних повинне покладатися на те, що отримання значень не відноситься до діяльності проектування (це інша сторона проєкційних дій).

Очевидно, що для вирішення цих проблем потрібно застосування нових методик та інструментів проектування, так як для абстрактної моделі даних, що володіє більш багатими структурними можливостями в порівнянні з реляційною моделлю, потрібно інший метод скорочення надмірності даних або іншими словами метод нормалізації схеми. При цьому загальновідомого методу нормалізації ООБД не існує, є тільки окремі методи, представлені в різних роботах.

Виходячи з вищевикладеного виникає необхідність дослідження і розробки методу і засобів проектування та контролю схеми ООБД для ІС, які дозволили б повною мірою розкрити всі сторони процесу нормалізації ООБД. При цьому важливо щоб цей метод ґрунтувався на теоретичній базі, головним елементом якої є модель даних, а схема була виконана на візуально-декларативній мові проектування. Актуальність дослідження обумовлена тим що, в даний час існує колосальний розрив між потенціалом ООБД і інструментальними засобами і методами їх проектування, які дозволили б зробити ООБД загальнодоступними і вивели б їх на новий етап розвитку.

Для усунення деяких недоліків, які властиві ООБД, зокрема відсутність універсальної моделі даних, яка є однією із суттєвих недоліків ООБД, необхідно реалізувати модель представлення структури інформаційного забезпечення, за допомогою якої необхідно вирішити наступні питання:

- проводити контроль правильності структури та взаємозв'язок між об'єктами уже проєктованих ООБД;
- проводити контроль правильності структури та взаємозв'язок між

об'єктами на етапі проектування ООБД;

– забезпечити автоматизоване проектування структури та взаємозв'язок між об'єктами ООБД.

Використовуючи математичний апарат теорії формальних граматики та теорії кінцевих автоматів запропоновано формалізовану модель опису структури ООБД. Структуру ООБД представимо як кінцеву множину вершин, для яких задано порядок проходження. Формальна математична модель структури ООБД представлена схемою структури, обумовленою як шістька:

$$S = (K, KE, KT, F, FO, KO),$$

де:

–  $K = \{K_{11} \dots K_{1S} \dots K_{21} \dots K_{2S} \dots K_{n1} \dots K_{nS} \dots\}$  – множина класів ООБД;

–  $KE = \{K_{e1} \dots K_{e1p}, K_{e2} \dots K_{e2p}, K_{en} \dots K_{npp}\}$  – множина екземплярів класів;

–  $F : K \times KE \rightarrow K$  – функція переходів, що реалізує відображення  $KE$  у  $K$ ; функція  $F$  деяким парам основний клас – вхідний екземпляр  $(K_{si}, K_{epi})$

ставить у відповідність деякий клас  $K_{si-1} = F(K_{si}, K_{epi})$ ,  $K_{si-1} \in K$ ;

–  $KT = \{K_t, \dots, K_{tp}, \dots, K_{tp}\}$  – множина таблиць ООБД;

–  $FO : K \times KT \rightarrow K$  – функція переходів, що реалізує  $KT$  у  $K$ ; функція  $FO$ , деяким парам основний клас – вхідна таблиця  $(K_{si}, K_{tpi})$  ставить у відповідність деякий клас  $K_{si-1} = FO(K_{si}, K_{tpi})$ ;

–  $KO \in K$  – початкова вершина БД.

У загальному вигляді схема ООБД представлена як  $CH = \{S_i\}_{i=1..T}$  – множина підмоделей структур БД, з яких складається ООБД. Структура  $S_1$ , використовується для ініціалізації процесу розпізнавання, і називається початковою. Множина таблиць  $T$  – це таблиці структури, що описується, які надходять на вхід програми-аналізатора безпосередньо з вхідного потоку. Кожному символу  $kt_p$  з множини  $KT$  відповідає своя структура  $S_i$ , що входить до складу схеми ООБД, тобто:

$$\forall kt_p \exists S_i (kt_p \in KT, S_i \in CH).$$

Аналіз структури об'єктно-орієнтованих баз даних виконується в 2 етапи:

I етап – аналізується загальна структура ООБД, на даному етапі аналізуються класи і екземпляри ООБД та зв'язки між ними:

$$kt_p \in KT \exists S_i \in CH.$$

II етап – детальний аналіз структури ООБД.

Аналіз структури ООБД еквівалентний побудові дерева розбору методом "нагору": на кожному кроці алгоритму будується один з рівнів у дереві розбору, "піднімаючись" від листів до кореня. Методика аналізу структури ООБД з використанням формальної математичної моделі полягає в наступному:

1) таблиці  $a_i$  розміщені на листках в дереві розбору, замінюються класом  $A$ , для якого в формальній моделі є функція  $A \rightarrow a$  (іншими словами, робиться "згортка" таблиці  $a$  до класу  $A$ );

2) потім багаторазово (доти, поки не буде досягнуто вершини ООБД) виконуються наступні кроки: отриманий на попередньому кроці клас  $A$  і розташована безпосередньо в даному класі чергова таблиця  $a_i$  замінюється класом  $B$ , для якого в формальній моделі є функція  $B \rightarrow Aa_i$  ( $i = 2, 3, \dots, n$ ).

При роботі даного алгоритму можливі наступні ситуації.

1) Прочитана вся структура ООБД, що аналізується; на кожному кроці знаходилася єдина потрібна "згортка"; на останньому кроці згортка відбулася до вершини  $K0$ . Це означає, що структура ООБД, яка аналізується, відповідає опису формальній математичній моделі структури ООБД.

2) Прочитана вся структура ООБД, що аналізується; на кожному кроці знаходилася єдина потрібна "згортка"; на останньому кроці згортка відбулася до символу, відмінному від  $K0$ . Це означає, що структура ООБД, яка аналізується, не відповідає опису формальної математичної моделі структури ООБД.

3) На деякому кроці не знайшлося потрібної згортки, тобто для отриманого на попередньому кроці класу  $K$  і розташованої безпосередньо в ньому таблиці  $kt_i$  не знайшлося класу  $B$ , для якого в правилах була би функція виводу  $B \rightarrow Kkt_i$ . Це означає, що структура ООБД, яка аналізується, не відповідає опису формальної математичної моделі структури ООБД.

4) На деякому кроці роботи виявилось, що є більше однієї придатної згортки, тобто в правилах різні класи мають правила виводу з однаковими правими частинами, і тому незрозуміло, до якого з них робити згортку. У даній ситуації необхідно робити згортку в усіх напрямках. Якщо хоча б один з напрямків приведе до символу, відмінному від  $K0$ , то це означає, що структура ООБД, яка аналізується, не відповідає опису формальної математичної моделі ООБД, у протилежному випадку відповідає.

Для того, щоб швидше знаходити функцію з необхідною правою частиною, необхідно зафіксувати всі можливі згортки (це визначається тільки правилами і не залежить від виду структури ООБД). Це можна зробити у вигляді таблиці, рядки якої позначені класами структури ООБД, стовпці – таблицями. Значення кожного елемента таблиці – це клас, до якого можна згорнути пари "клас–таблиця", якими позначені відповідні рядок і стовбець.

Отже, розробка методів та засобів на основі запропонованої формальної моделі ООБД дозволить вирішити наступні задачі:

- проводити контроль правильності структури та взаємозв'язок між об'єктами уже проєктованих ООБД;
- проводити контроль правильності структури та взаємозв'язок між об'єктами на етапі проєктування ООБД;
- забезпечити автоматизоване проєктування структури та взаємозв'язок між об'єктами ООБД.

- мінімізувати чи усунути проблеми цілісності даних;
- підвищити загальну продуктивність інформаційної системи;
- зменшити вартість розробки і супроводу бази даних.

#### Література

1. Курейчик В.М., Гладков Л.А., Курейчик В.В. Дискретная математика. Теория множеств. Учебное пособие. 4.1. — Таганрог: Изд-во ТРТУ, 2005.
2. Джулій В.М. Представлення моделі ООБД інформаційного забезпечення тестового діагностування мікропроцесорних пристроїв / Джулій В. М., Муляр І. В. // Вісник ТУП. – 2004. – №2. Частина 1, том 1.– С.79 -83.
3. Джулій В.М. Гіпертекстова модель представлення інформації в базах діагностичних даних / Джулій В.М., Муляр І.В. // Вісник ТУП. – 2001. – №1. – С.189 -191.

### **Архітектура програмного комплексу забезпечення безпеки виявлення і протидії DDoS-атакам**

Савіцька О.О., Джулій В.М., Муляр І.В.  
Хмельницький національний університет

Програмний комплекс виявлення початку атаки в режимі реального часу розраховує середньоквадратичне відхилення з урахуванням актуальних сезонних періодів за кількістю запитів до мережного ресурсу в кожному періоді. Програмний комплекс дає можливість задати розмірність розглянутих періодів: 1 хвилина, 15 хвилин, 1 годину і т.д. А також вести моніторинг відразу по декількох періодах. На підставі розрахованого середньоквадратичного відхилення задається верхня межа кількості запитів до мережного ресурсу відповідного періоду.

Програмний комплекс виявлення початку атаки має гнучкі налаштування, що дозволяють задати чутливість до можливої атаки (лістинг 1). Конфігураційні дані виділені в окремий php-файл, що дає додаткові можливості як з точки зору зручності, так і з точки зору безпеки. Чутливість варіюється за допомогою корекції границі, а також порушенням границі відразу в декількох періодах різного розміру. Наприклад, при порушенні границі на хвилинних інтервалах засіб може тільки сповістити зацікавлених осіб про збільшення активності. У разі порушення границі також на п'ятихвилинному інтервалі відбувається повне включення механізму захисту.

Лістинг 1 - Фрагмент конфігураційного файлу

```
//час періоду мережевої активності в секундах, 86400 добу, 604800 тиждень
```

```
$Loop = 40400;
```

```
//період для дослідження в секундах
```