

УДК 378.147

М. С. Сафонов

МЕТОД РЕІНЖІНІРИНГУ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ОБ'ЄКТІВ УПРАВЛІННЯ

***Анотація.** Проведено аналіз існуючої інформаційної системи Херсонського політехнічного коледжу Одеського національного політехнічного університету. Визначені основні недоліки її організації. Розглянуті методи реінжинірингу цієї системи з метою вдосконалення процесу руху потоків даних між підсистемами. Застосовано об'єктно-орієнтований реінжиніринг для вирішення задачі.*

***Ключові слова:** інформаційна система, реінжиніринг, потоки даних, об'єктно-орієнтований метод, підсистема, рух потоків*

M. Safonov

METHOD OF REENGINEERING OF AN INFORMATION SYSTEM WITH USE OF OBJECTS OF MANAGEMENT

***Abstract.** The analysis of an existing information system of the Kherson polytechnical college of the Odessa national polytechnical university is carried out. Main limitation of its organization is defined. Methods of reengineering of the given system, for the purpose of process improvement of driving of data streams between subsystems are considered. Application of object-oriented reengineering for the task decision.*

***Keywords:** Information system, reengineering, databases, object-oriented method*

М. С. Сафонов

МЕТОД РЕІНЖІНІРИНГА ІНФОРМАЦІЙНОЇ СИСТЕМИ С ІСПОЛЬЗОВАННЯМ ОБ'ЄКТІВ УПРАВЛІННЯ

***Аннотация.** Проведен анализ существующей информационной системы Херсонского политехнического колледжа Одесского национального политехнического университета. Определены основные недостатки ее организации. Рассмотрены методы реинжиниринга этой системы с целью усовершенствования процесса движения потоков данных между подсистемами. Применяется объектно-ориентированный реинжиниринг для решения задачи.*

***Ключевые слова:** информационная система, реинжиниринг, потоки данных, объектно-ориентированный метод, подсистема, движение потоков*

Сьогодні можна казати, що ера, коли розроблювачі інформаційних систем (ІС) приходили в організацію й починали проекти інформатизації «з нуля», минула. Настає час проектів систематичної трансформації існуючих ІС або ера реінжинірингу ІС.

Відокремимо такий клас ІС, які відповідають певному переліку тверджень:

всі структурні підрозділи мають власні автоматизовані системи управління (АСУ), які виконують визначені завдання і не виходять за межі своєї бази даних (БД);

кожна з цих БД може містити однакову інформацію;

база даних в ІС мають кілька різних архітектур, несумісних одна з одною;

вихідні коди програм частково або повністю відсутні, що виключає можливість безпосередньої їх зміни;

існує повний доступ до всіх БД та відома їх структура.

© Сафонов М.С., 2014

Завданнями реінжинірингу виступають: об'єднання інформаційних ресурсів структурних підрозділів;

створення інтегрованої ІС управління, функціонуючою в реальному масштабі часу; оператівне та автоматичне отримання актуальних даних;

загальне зниження витрат часу на роботу з системою;

можливість гнучкого реагування на зміни вимог до системи.

Описаним вище властивостям відповідає ІС Херсонського політехнічного коледжу Одеського національного політехнічного університету (ХПТК ОНПУ). Дослідження було проведено на базі наступних АСУ.

- Бібліотека.
- Методичний кабінет.
- Педнавантаження.
- Соціальний педагог.
- Відділ кадрів.
- Приймальна комісія.

- Архів.
- Тестування знань.
- Канцелярія.

Автоматизовані системи управління «Бібліотека» організовує збір інформації по книжковому фонду. База даних АСУ «Бібліотека» зберігає дані про читачів (викладачів та студентів), які мають книжки у себе на руках. Рух книжки прослідковується за її інвентарним номером. БД працює під управлінням СУБД MySQL 5.0.

Автоматизовані системи управління «Методичний кабінет» організовує систематизацію інформації щодо наявності та актуальності методичного забезпечення в електронній базі коледжу. В БД по кожному викладачу ведеться інформація про кількість розробленого методичного матеріалу (лекцій, практичних, лабораторних, самостійних, модульних робіт) з певної дисципліни. БД працює під управлінням СУБД MySQL 5.0.

Автоматизовані системи управління «Педнавантаження» автоматизує формування навантаження по кожному викладачу чи групі згідно з навчальним планом. В БД зберігається інформація по навантаженню викладачів, дисциплінах, які викладаються, керівництва групою. БД працює під управлінням СУБД Borland Interbase 7.1.

Автоматизовані системи управління «Соціальний педагог» автоматизує ведення інформації про студентів. БД вміщує дані про соціальне положення, поведінку, догани, стан здоров'я тощо. БД працює під управлінням СУБД MySQL 5.0.

Автоматизовані системи управління «Відділ кадрів» автоматизує ведення інформації про співробітників та студентів коледжу. По кожному співробітнику в БД зберігаються дані про посади, які він займав, дати прийняття на роботу та переведення на іншу посаду, дані про освіту. Також БД зберігає інформацію по студентам: вік, рік ступу, група, форми навчання та фінансування. БД працює під управлінням СУБД Borland Interbase 7.1.

Автоматизовані системи управління «Приймальна комісія» організовує збір інформації по кожному вступнику: контактні дані, освіта, спеціальність для вступу, отрима-

ний на іспиті бал. БД працює під управлінням СУБД MySQL 5.0.

Автоматизовані системи управління «Архів» автоматизує систематизацію документів з тривалим строком зберігання: накази, дипломи та інше. БД працює під управлінням СУБД Borland Interbase 7.1.

Автоматизовані системи управління «Тестування знань» автоматизує процес проведення модульних та семестрових контролів. В БД зберігається перелік студентів по групам, які проходили тестування та отримані ними бали по дисциплінам. База даних працює під управлінням СУБД Borland Interbase 7.1.

Автоматизовані системи управління «Канцелярія» виконує автоматизацію ведення документооберту (наказів, відряджень, відпусток тощо). БД працює під управлінням СУБД MySQL 5.0.

В даній ІС не організовано рух потоків даних від однієї АСУ до іншої. А це є суттєвим недоліком, тому що всі вони потребують своєчасного отримання даних для організації контролю та управління ІС в цілому.

Розглянута ІС характеризується недоліками організації інформаційних потоків. Серед типових недоліків системи слід назвати:

дублювання інформації, що надається (як правило однакову інформацію кожного разу доводиться вводити вручну, що займає багато часу);

невчасність надання інформації;

інформація може не доходити до адресата;

після здобуття інформації можуть бути потрібні уточнення від інших підрозділів.

Важливим завданням даної роботи стає вдосконалення системи інформаційних потоків, зміна алгоритмів проходження даних, автоматизація передачі інформації, тобто її реінжиніринг.

Система управління інформаційними потоками повинна органічно вбудовуватися в загальну систему інформаційних потоків учбового закладу, інакше станеться відторгнення співробітниками нав'язаної ним схеми.

На рис. 1 представлена схема направленості потоків інформації в системі на прикладі студента та викладача в ІС ХПТК ОНПУ.

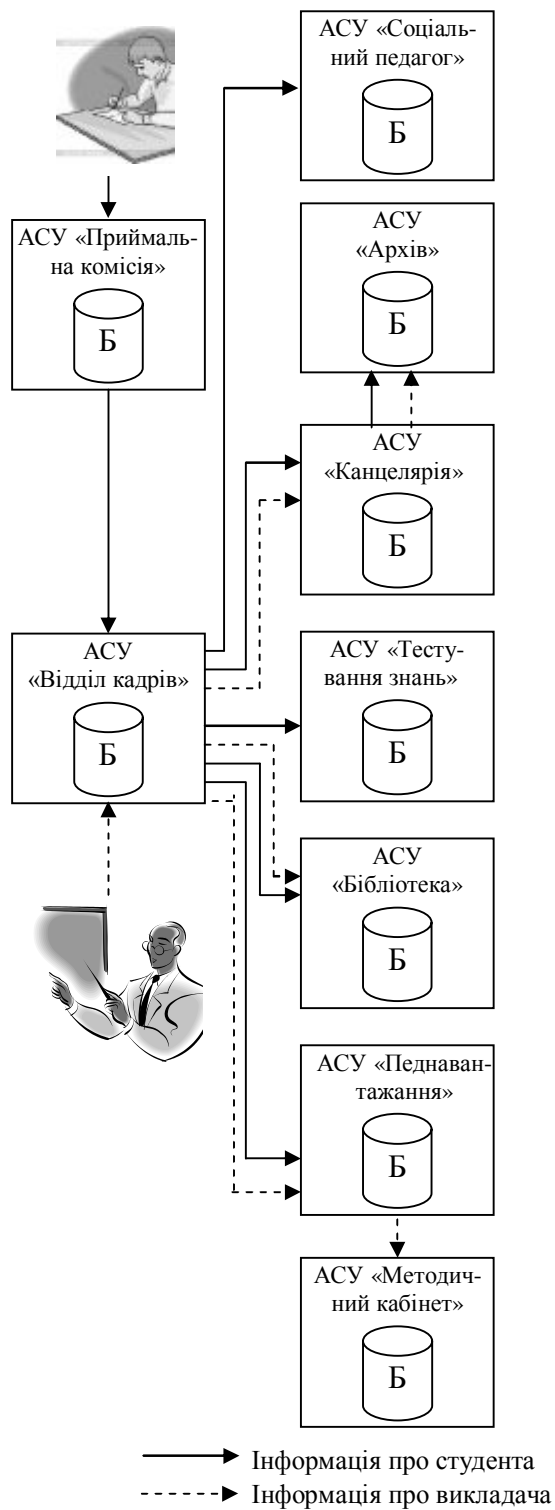


Рис. 1. Рух потоків інформації про студента та викладача

Розглянемо найбільш поширені методи реінжинірингу систем та можливість їх використання в рамках описаної ІС.

Забезпечуючи концептуальне розуміння процесу реінжинірингу ІС, у ряді робіт [1, 2]

визначаються основні види діяльності (фази), що співвідносяться із цим процесом.

Так, в [1] розглядаються наступні основні фази:

оцінки показників проекту по реінжинірингу, у тому числі характеристик успадкованої інформаційної системи (фаза оцінки);

аналізу розв'язків по реінжинірингу, у тому числі ухвалення рішення про необхідність проведення робіт з реінжинірингу або супроводу ІС;

здійснення реінжинірингу (виконання робіт з реінжинірингу);

впровадження системи, трансформованої в результаті проведення реінжинірингу.

Даний підхід описує лише загальні положення та не описує конкретних кроків щодо вирішення завдання реінжинірингу систем з практичної точки зору.

Інший підхід до визначення діяльності по реінжинірингу базується на так званій моделі «підкови» [2, 3]. В її основу покладені наступні процеси, що співвідносяться з реінжинірингом ІС:

аналіз існуючої системи, заснований на одному або декількох її логічних описів;

трансформація цих логічних описів у новий, поліпшений логічний опис системи.

розробка нової системи, заснованої на нових логічних описах системи.

До недоліків методу «підкови» можна також віднести:

відсутність гарантії (і навіть переконаності), що шляхом модифікації вдасться привести інформаційну систему у відповідність новим вимогам;

складність прогнозування процесу, оцінки необхідних ресурсів;

складність структурної моделі, що описує реінжиніринг інформаційної системи;

досить високу вірогідність виникнення «хвилі змін» (модифікація одного компонента вимагає модифікації інших компонентів, що взаємодіють з ним, ті у свою чергу також вимагають модифікації зв'язаних компонентів і таким чином кількість компонентів лавиноподібно зростає).

Модель «підкови» є найпоширенішою у вирішенні питань з реінжинірингу, але вона не розрахована на реінжиніринг систем, які

не подлягають втручанню у код програми. Що робить неможливим її використання..

Підхід, запропонований в [4–6], дуже близький до підходу, заснованого на моделі «підкова». Характеризуючи життєвий цикл реінжинірингу ІС, автори Сандер Т. и Дукас Р. визначають наступні кроки процесу об'єктно-орієнтованого реінжинірингу:

аналіз вимог для виявлення конкретних цілей реінжинірингу успадкованої системи;

відновлення моделі, у тому числі документування й розуміння структури успадкованої системи;

виявлення проблем, пов'язаних з успадкованою системою;

аналіз проблем, що включає вибір архітектури, що дозволяє усунути виявлені в успадкованій системі дефекти;

реорганізація, що включає вибір і застосування оптимального підходу трансформації успадкованої системи;

поширення змін.

В основу підходу, що пропонується авторами [4, 5], покладена декомпозиція структури системи на компоненти використаного інтерфейсу, компоненти-додатка й компоненти-управління базами даних. Кожний структурний елемент системи розглядається як окремий об'єкт, який має певні властивості, реагує на події в системі та його можливо віднести до певного класу об'єктів. Редагування коду не є можливим, але існує повний доступ до БД, тому в даних умовах саме об'єктно-орієнтований підхід є найбільш придатним для реінжинірингу даної існуючої ІС.

Виділимо головні недоліки всіх вище перерахованих методів реінжинірингу:

неможливість проведення реінжинірингу без втручання в програмний код;

значні затрати часу при реінжинірингу складної інформаційної системи, що значно збільшує вартість проведення робіт;

значна складність сполучення декількох систем, що були написані на різних мовах програмування та використовують БД різних архітектур, в єдину цілісну інформаційну систему.

У зв'язку з цим були поставлені наступні завданнями даного дослідження:

розробка методу реінжинірингу інформаційної системи, який би дозволяв змінювати та контролювати потоки даних без втручання в програмний код;

даний метод повинен бути досить гнучким в управлінні, що дозволить зменшувати чи збільшувати навантаження на робочі станції з метою покращення часових характеристик;

розробка інформаційної моделі методу реінжинірингу, яка б в реальному часі представляла інформацію про швидкісні характеристики роботи системи;

метод не повинен вимагати від адміністратора інформаційної системи знань великої кількості мов програмування, а лише основи алгоритмізації та досвід роботи з БД.

Так як БД в досліджуваній ІС мають відмінну архітектуру, то рух даних повинен відбуватися за допомогою універсальної комп'ютерної мови, що підтримується у всіх реляційних базах даних – SQL. При всіх своїх змінах, SQL залишається єдиним механізмом зв'язку між прикладним програмним забезпеченням і базою даних. Кожна команда SQL – це або запит даних з бази, або звертання до бази даних, яке приводить до зміни даних у базі.

Найчастіше різні БД розміщені на різних робочих станціях. При цьому має сенс розподілити частину часу обробки на виділені комп'ютери. На кожному ПК варто розмістити спеціальні об'єкти управління, які б брали на себе операції перебору, а також прийому вхідної інформації й передачі вихідної основній системі управління.

Всі об'єкти управління взаємодіють безпосередньо з БД під управління головної системи (ГСУ), в обхід програмній частині, що дозволяє використовувати готові програмні вирішення стороннього розробника без втручання в програмний код (рис. 2).

Згідно з даним методом основна система відсилає команду першому об'єкту управління й майже у той же час відсилає її другому й так далі. Після відсилання всіх команд відбувається збір інформації з тих же робочих станцій або БД.

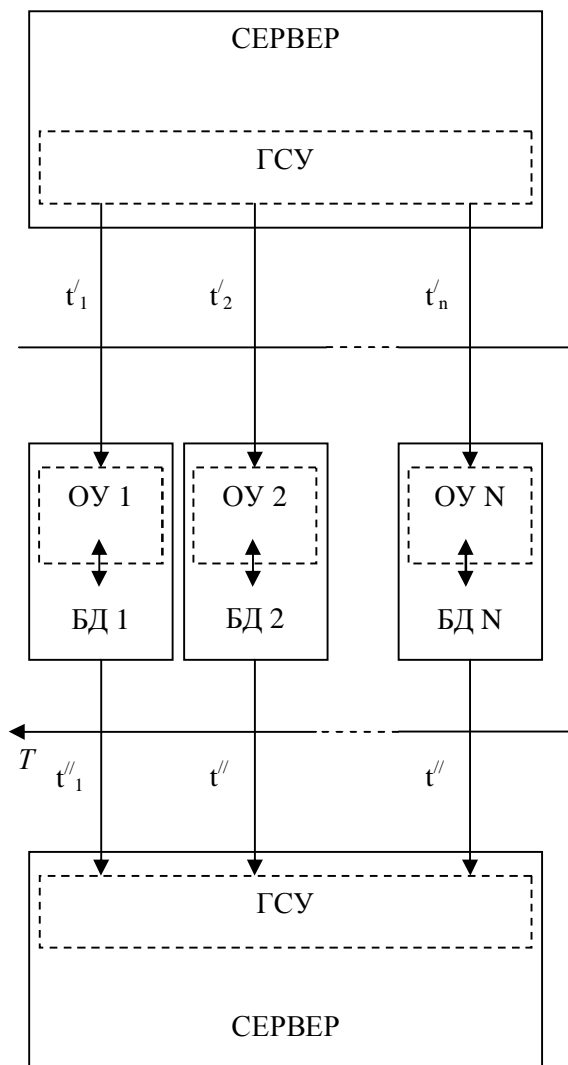


Рис. 2. Загальна схема метода з використанням об'єктів управління

Не виключена ситуація, коли та або інша робоча станція буде не готова надати відповідь у призначений строк. І тоді ГСУ прийдеться очікувати якийсь час Δt своєї бездії. У такий спосіб відбувається збільшення часу $T_{\text{бд}}$ на деяку величину Δt . І завдання полягає в тому, щоб максимально зменшити її значення ($\Delta t \rightarrow 0$):

$$T_{\text{аа}} = \sum_{i=1}^n (t_i + t'_i + t''_i) + \Delta t. \quad (1)$$

У зв'язку із цим доцільно ввести коефіцієнт λ , що буде відображати пріоритет і послідовність відсилання команд й одержань відповідей. Тобто БД із найбільшим розміром і часом обробки буде мати найвищий пріоритет ($\lambda = 1$) і оброблятися першою. Відповідно БД із найменшим розміром і ча-

сом обробки буде мати найменший пріоритет ($\lambda \rightarrow n$) [7]:

$$\lambda_1 < \lambda_2 < \dots < \lambda_n \Rightarrow t_1 > t_2 > \dots > t_n. \quad (2)$$

Окрім пріоритету отримання команд λ , потрібно ввести пріоритети роботи об'єктів управління:

- пріоритет швидкості, в процесі роботи якого об'єкти управління після надходження команди негайно перейдуть до її виконання;
- пріоритет відмовостійкості, в процесі роботи якого об'єкти управління після надходження команди будуть аналізувати поточний рівень навантаження робочої станції, і у випадку його зниження перейдуть до виконання.

Саме встановлення пріоритетів роботи системи робить її досить гнучкою [8, 9].

Побудуємо тривимірну модель роботи об'єктно-орієнтовного метода управління потоками даних в БД (рис. 3).

В якості тривимірної моделі візьмемо просторову спіраль. Для і станів параметрами побудови даної моделі будуть виступати радіус спіралі (час обробки однієї команди) t'_i , крок витка (час між відправленнями команд різним об'єктам управління) t_i , кількість витків m , що відповідає кількості ітерацій системи, кількість об'єктів управління n та висота спіралі $T_{\text{бд}}$.

Середній час $t_{\text{сер}}$ від надсилання команди об'єкту управління до отримання від нього відповіді можна визначити як відношення загального часу обробки $T_{\text{бд}}$ до кількості повторюваних ітерацій m :

$$t_{\text{сер}} = \frac{T_{\text{бд}}}{m}. \quad (3)$$

Якщо кожний виток відповідає одній ітерації системи, то при $i \in [1..m \cdot n]$ визначимо модель аналітично:

$$\begin{cases} x_i = t'_{ni} \cdot \cos\left(\frac{\pi \cdot i}{n}\right) \\ y_i = t'_{ni} \cdot \sin\left(\frac{\pi \cdot i}{n}\right) \\ z_i = \frac{T_{\text{аа}} \cdot i}{n \cdot m} \end{cases}. \quad (4)$$

Основною характеристикою роботи інформаційної системи в цілому згідно з даною моделлю буде виступати довжина побудованої тривимірної спіралі L [10].

Тобто зміна будь яких часових характеристик спричинить автоматичну перебудову моделі з урахуванням мінімальної довжини спірального шляху.

Для знаходження довжини спірального шляху визначимо систему диференціальних рівнянь:

$$\begin{cases} x'_i = -t'_{ni} \cdot \sin\left(\frac{\pi \cdot i}{n}\right) \\ y'_i = t'_{ni} \cdot \cos\left(\frac{\pi \cdot i}{n}\right) \\ z'_i = \frac{T_{\omega\omega}}{m} \end{cases} \quad (5)$$

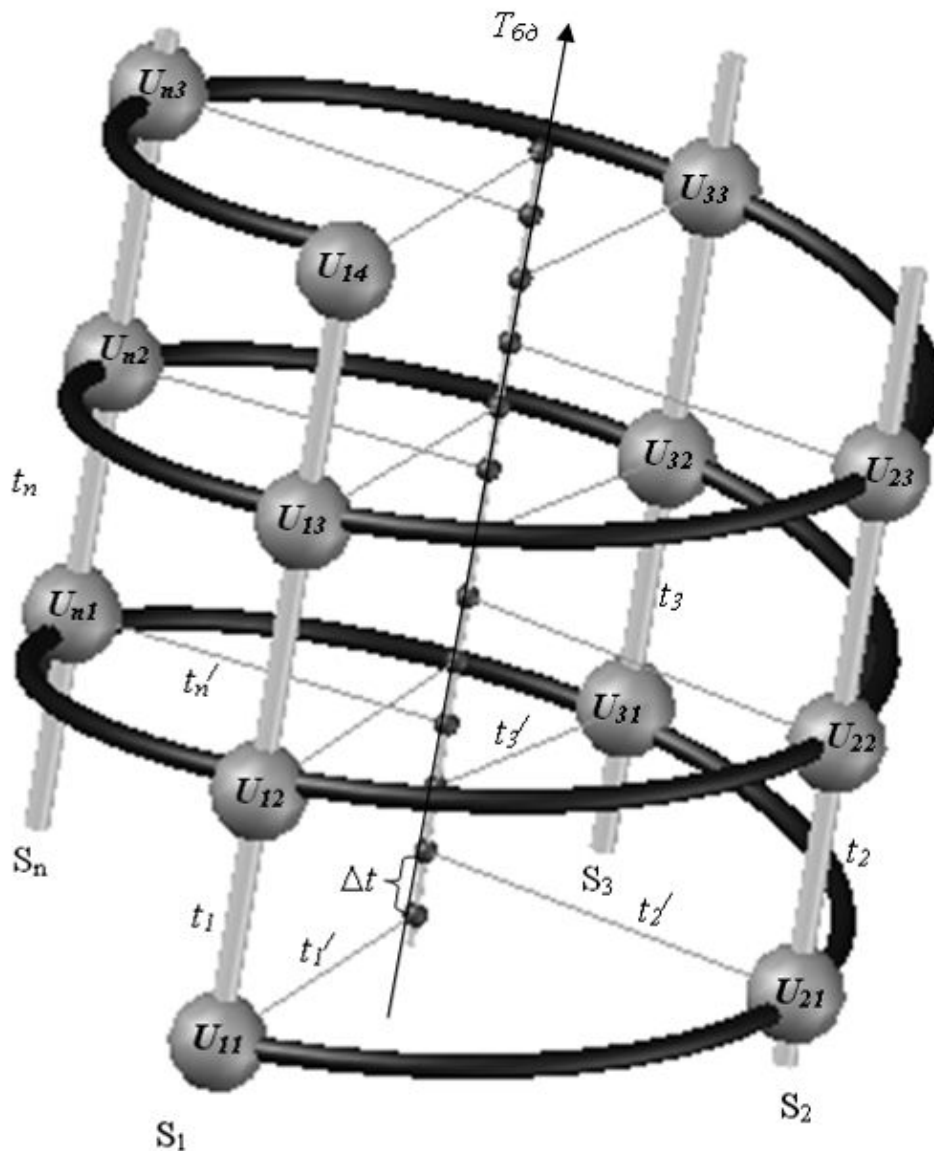


Рис. 3 Тривимірна модель об'єкто-орієнтованого метода управління потоками інформації в БД.

Якщо $i \in [0, n]$, то $\frac{\pi \cdot i}{n} \in [0, \pi]$. Визначимо довжину спірального шляху L

$$L = \int_0^{\pi} \sqrt{\left(-t_{ni}' \cdot \sin\left(\frac{\pi \cdot i}{n}\right)\right)^2 + \left(t_{ni}' \cdot \cos\left(\frac{\pi \cdot i}{n}\right)\right)^2 + \left(\frac{T_{\delta\delta}}{m}\right)^2} d\left(\frac{\pi \cdot i}{n}\right). \quad (6)$$

В формулі (6) зробимо заміну виразу $\frac{\pi \cdot i}{n}$ на змінну x :

$$L = \int_0^{\pi} \sqrt{\left(-t_{ni}' \cdot \sin(x)\right)^2 + \left(t_{ni}' \cdot \cos(x)\right)^2 + \left(\frac{T_{\delta\delta}}{m}\right)^2} d(x). \quad (7)$$

Спростимо останній вираз:

$$L = \int_0^{\pi} \sqrt{\left(t_{ni}'\right)^2 \cdot (\sin(x)^2 + \cos(x)^2) + \left(\frac{T_{\delta\delta}}{m}\right)^2} d(x) = \pi \sqrt{\left(t_{ni}'\right)^2 + \left(\frac{T_{\delta\delta}}{m}\right)^2}. \quad (8)$$

Представлений вираз (8) дозволяє розрахувати загальну довжину спірального шляху L , який характеризує часові параметри роботи об'єктно-орієнтованого методу управління потоками інформації в мережевих БД.

Довжина спірального шляху здебільшого залежить від часу обробки запиту на робочій станції об'єктом управління (t_i') та часу переходу управління від однієї станції до іншої (Δt_i). Саме введення пріоритетів роботи системи вплине на часові характеристики роботи системи. Тобто в процесі формування моделі об'єктно-орієнтованого методу управління потоками інформації в БД будуть визначатися ділянки з часовими характеристиками, що значно більше за інші. Це буде означати про доцільність зміни поточного пріоритету роботи об'єкта управління. За допомогою виразу (8) можливо оперативно визначати характеристики стану інформаційної системи та своєчасно виявити причини затримки потоків даних, за рахунок ведення накопичуваної бази знань попередніх результатів роботи методу.

В процесі впровадження даної системи було розроблено програмне забезпечення, написане на мові програмування Delphi 7, яке складається з двох частин: об'єкту управління та ГСУ. Об'єкти управління з пріоритетом відмовостійкості були розміщені на дев'ятих робочих станціях, де знаходяться цільові БД, а ГСУ – на сервері. З

головного комп'ютера було відправлено по три простих SQL-запита. В процесі автоматичної побудови моделі розрахована довжина спірального шляху, яка дорівнювала 42,91 с. Після зміни пріоритету роботи чотирьох з дев'яти об'єктів управління на пріоритет швидкості та відправки до кожного з них по три SQL-запита, була перебудована модель, яка показала зменшення спірального шляху до 40,34 с. Переведення всіх дев'яти об'єктів управління в режим пріоритету швидкості не показав істотних зменшень часу обробки запитів, що вказує на ключову роль саме перших чотирьох станцій. Так за допомогою об'єктно-орієнтованої моделі та виразу (8) через зміну пріоритетів роботи різних об'єктів управління був зменшений час обробки запитів ГСУ до робочих станцій на 6 %.

Даний метод дозволив перенаправити інформаційні потоки в різні БД без зміни програмних кодів та знань багатьох мов програмування. Замість редагування дев'ятих програм була написана одна, яка зв'язувала роботу всіх систем в єдине ціле.

Реінжиніринг з використанням об'єктів управління в повній мірі вирішує завдання по вдосконаленню роботи інформаційної системи без втручання в вихідний код програм та повністю задовольняє потребам роботи ІС ХПТК ОНПУ. Кожний об'єкт управління за допомогою ГСУ виконує роботи по перене-

сенню та аналізу інформації з однієї БД в іншу, навіть якщо вони мають різну архітектуру.

Даний метод реінжинірингу доцільно використовувати у тих випадках, коли:

використовується готова інформаційна система стороннього розробника, який регулярно випускає нові версії ІС із відкритою структурою БД;

стратегічні плани організації не визначені повністю (можливий перехід на іншу платформу або використовується декілька платформ);

бюджет організації досить обмежений (немає можливості прийняти висококваліфікованих фахівців для проведення реінжиніринга).

Список використаної літератури

1. John Bergey, Reengineering Process Framework, Software Engineering [Text] / John Bergey, William Hefley, Walter Lamia, Dennis Smith A // Institute, Carnegie Mellon University, Pittsburgh, 1995, pp. 92 – 96.

2. John Bergey, Options Analysis for Reengineering (OAR): Issues and Conceptual Approach [Text] / John Bergey, Dennis Smith, Nelson Weiderman, Steven Woods // Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1999, pp. 67 – 74.

3. Carriere S. J., Software Architectural Transformation [Text] / S. J. Carriere, S. Woods, R. Kazman // Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1999, pp. 121 – 130.

4. Sander T. Modeling Object-Oriented Software for Reverse Engineering and Refactoring [Text] / T. Sander // Thesis, University of Bern, Bern, Switzerland, 2001, pp. 142 – 149.

5. Ducasse, S. Rretro-Conception d'Application `a Objets Reengineering Object-Oriented Applications [Text] / S. Ducasse // Universitre Pierre et Marie Curie, Paris, 2001, pp. 81 – 93 .

6. Holger B. The FAMOOS Object-Oriented Reengineering Handbook [Electronic resource] Access mode: /<http://www.iam.unibe.ch/~famoos/handbook/> (12.11.2013).

7. Сафонов М. С., Прогнозування стану показників об'єктно-орієнтованої моделі в інформаційній системі [Текст] / М. С. Сафонов, О. Є. Яковенко // Збірник наукових праць / Інформаційні технології в освіті, науці та виробництві – Одеса : – 2013. – Вип. 3(4) – С. 92 – 98.

8. Каталевский Д. Ю. Основы имитационного моделирования и системного анализа в управлении [Текст] / Д. Ю. Каталевский – М. : Издательство Московского университета. – 2011. – 304 с. ISBN 978-5-211-05923-8.

9. Основные модели информационных систем и способы увеличения их эффективности [Электронный ресурс] – Режим доступа: <http://www.e-lines.ru/info/63-modeli-informacionnyh-system> (13.02.13).

10. Сафонов М. С. Моделирование управления потоками данных в информационной системе [Текст] / М. С. Сафонов, А. Е. Яковенко // Труды Одесского политехнического университета. – Одесса. – 2013. – Вип. 1, – С. 97 – 103.

Отримано 07.03.2014

References

1. Bergey John, Hefley William, Lamia Walter, and Smith A. Dennis. Reengineering Process Framework, (1995), *Software Engineering. Institute, Carnegie Mellon University, Pittsburgh*, pp. 92 – 96.

2. Bergey John, Smith Dennis, Weiderman Nelson, and Woods Steven. Options Analysis for Reengineering (OAR): Issues and Conceptual Approach, (1999), *Software Engineering Institute, Carnegie Mellon University, Pittsburgh*, pp. 67 – 74.

3. Carriere S.J., Woods S., and Kazman R. Software Architectural Transformation, (1999), *Software Engineering Institute, Carnegie Mellon University, Pittsburgh*, pp. 121 – 130.

4. Sander T. Modeling Object-Oriented Software for Reverse Engineering and Refactoring. Thesis, (2001), *University of Bern, Bern, Switzerland*, pp. 142 – 149.

5. Ducasse S. Rretro-Conception D'application `a Objets Reengineering Object-Oriented Applications, (2001), *Universitre Pierre et Marie Curie, Paris*, pp. 81 – 93 .

6. Holger B. The FAMOOS Object-Oriented Reengineering Handbook [Electronic resource]. Access mode: <http://www.iam.unibe.ch/~famoos/handbook/> (12.11.2013).

7. Safonov M.S., and Yakovenko A.E. Prognozuvanna stanu pokaznikov obektno-orientirovannoi modeli v informatiinij sistemi. [The Forecast of a Condition of Indicators of Object-oriented Model in Information System], (2013), *Informationnie Tehnologii v Obrazovanii, Nauke i Proizvodstve*, Odessa, Ukraine, 3(4), pp. 92 – 98 (in Ukrainian).

8. Katalevsky D.J. Osnovi imitacionnogo modelirovania I sistemnogo analiza v upravlenii. [Bases of Imitating Modelling and the System Analysis in Management], (2011), Moscow, Russian Federation, *Izdatelstvo Moskovskogo Universiteta*,. 304 p. (in Russian). ISBN 978-5-211-05923-87.

9. Osnovnie modeli informacionnih system I sposobi uvelichenia ih effektivnosti. [The Basic Models of Information Systems and ways of Increase in their Efficiency.] [Electronic resource] (in Russian). Access mode: www.e-lines.ru/info/63-modeli-informacionnyh-system (13.02.2013).

10. Safonov M.S., and Yakovenko A.E. Modelirovanie upravlenia potokami danih v informacionnoi sisteme. [Modelling of Management by Data Flows in Information System], (2013), *Trudi Odesskogo Politehnicheskogo Uuniversiteta Publ.*, Odessa, Ukraine, pp. 97 – 103 (in Ukrainian).



Сафонов Михайло Сергійович, завідувач лабораторії Херсонського політехн. коледжу Одеського нац. політехн. ун-ту, м. Херсон, вул. 40 років Жовтня, 23, м/т 0662458185, e-mail: <saifonov_ms@mail.ru>