

тактику лікування і впливу на ці фактори. Крім ФР попереднього року, додаються й нові, які також підлягають аналізу й впливу лікувальних заходів.

### Висновки

Описано алгоритм підтримки прийняття рішень на основі інтелектуального аналізу даних, що лежить в основі медичної інформаційної системи. Алгоритм дозволяє виявити пацієнтів з високим ризиком розвитку ускладнення та є ефективним інструментом для лікаря при аналізі та корекції тактики лікування. Метою розробки алгоритму підтримки прийняття рішень для медичної інформаційної системи є зниження ризику розвитку ускладнень в ранньому післяопераційному періоді у хворих з ішемічною хворобою серця після аортокоронарного шунтування.

### Список використаних джерел

1. Кузнецов С. Д. Основы баз данных / С. Д. Кузнецов – 2-е изд. – М.: Интернет – Университет Информационных Технологий; БИНОМ. Лаборатория знаний. – 2007. – 484 с.
2. Дюк, В. Data Mining: учебный курс / В. Дюк, А. Самойленко – СПб. : «Питер». – 2001. Dyuck, V. Data Mining: uchebniy kurs / V. Dyuck, A. Samoilenko – SPb. : «Piter». – 2001.
3. Ланг, Т. А. Как описывать статистику в медицине / Руководство для авторов, редакторов и рецензентов. / Т. А. Ланг, М. Сесик – М. : Практическая Медицина, 2011. – 480 с.
4. Логистическая регрессия. Многомерные методы статистического анализа категориальных данных медицинских исследований : Уч. пособие / С. Г. Григорьев, В. И. Юнкеров, Н. Б. Клименко – СПб, 2001. – с. 10–21.
5. Ким, Дж.-О. Дискриминантный анализ // Факторный, дискриминантный и кластерный анализ / Дж.-О. Ким, Ч.У. Мьюллер, У.Р. Клекка, и др. – М. : Финансы и статистика, 1989. – с. 78–138.
6. Jakulin, A. Quantifying and Visualizing Attribute Interactions / A. Jakulin, I. Bratko // An Approach Based on Entropy. PKDD. – 2004. – V. 3. – P. 229–240.
7. Бирман, Э.Г. Сравнительный анализ методов прогнозирования / Э. Г. Бирман – НТИ. Сер. 2. – 1986. – № 1. – с. 11–16.
8. Яковенко А.В. Выявление структуры факторов риска развития острой сердечной недостаточности в раннем послеоперационном периоде / А.В. Яковенко, А.В. Руденко, Е.А. Настенко, Н.Л. Руденко, В.А. Павлов // Восточно-европейский журнал передовых технологий. – 2013. – № 3/10 (63). – с. 4–8.

УДК 004.45

## КОНТРОЛЬ ВЕРСИЙ ДИНАМИЧЕСКИ ПОДСОЕДИНЯЕМЫХ БИБЛИОТЕК В СИСТЕМЕ УПРАВЛЕНИЯ ИТ-ИНФРАСТРУКТУРОЙ

Ролик А.И.<sup>1)</sup>, Март Б.А.<sup>2)</sup>, Ступак Б.В.<sup>3)</sup>, Моргун Н.В.<sup>4)</sup>

*НТУУ «Киевский политехнический институт»*

*<sup>1)</sup> д.т.н., доцент; <sup>2)</sup> аспирант; <sup>3-4)</sup> студенты*

### І. Введение

Поддержание качества выполнения бизнес-процессов на стабильно высоком уровне обеспечивается системами управления ИТ-инфраструктурой (СУИ), которые совершают мониторинг актуального состояния и управление корпоративной ИТ-инфраструктурой (ИТС). Мониторинг и управляющее воздействие на объект управления осуществляется с помощью предложенной в [1] и реализованной в СУИ Smartbase ITS-Control, разработанной на базе НТУУ «КПИ», DALLF (Dynamic Auto Link Library Function), использующей рефлексии типов.

Механизм DALL имеет гибкий формат, позволяющий реализовать в DALL-библиотеках как мониторинг, так и управление. Согласно этому механизму общение с библиотекой производится при помощи определенной входной функции, имеющей набор входных и выходных параметров. Стандартные механизмы опроса на основе WMI и SNMP также реализованы в виде DALL-библиотек.

Достоинством такого механизма построения функций мониторинга/управления является возможность привлечения к расширению функционала разработчиков, не знакомых со структурой СУИ.

DALL-функции подсоединяются к СУИ в виде скомпилированных .dll файлов. Исходный код DALL-функций обладает специальной структурой, позволяющей выделить множество входных и выходных параметров, а так же метод, отвечающий за выполнение функции. Поскольку DALL-функции, предложенные сторонними разработчиками, не всегда будут обеспечивать штатный режим работы системы, возникает необходимость в модуле контроля версий DALL-библиотек, позволяющем администратору ИТС, вручную выбирать тот набор и те версии DALL-библиотек, которые лучше всего отвечают нуждам компании.

## II. Задача наибольшей общей последовательности

Очевидно, что загружать новые версии целиком в базу данных не является оптимальным ни с точки зрения хранения данных на жестком диске, ни по использованию пропускной способности канала передачи данных. Поэтому рациональным решением является хранение в БД отличий текущей версии DALL-функции от предыдущей.

В основу алгоритма нахождения разницы между двумя версиями файлов положено решение задачи нахождения наибольшей общей последовательности (НОП).

При постоянном количестве входных последовательностей решение этой задачи за полиномиальное время достигается средствами динамического программирования [2].

Предлагаемый алгоритм решения данной задачи базируется на следующих свойствах:

-  $НОП(X_n, Y_m) = (НОП(X_{n-1}, Y_{m-1}), x_n)$ , т.е. если две последовательности оканчиваются одним и тем же элементом, то для нахождения НОП достаточно найти НОП от подпоследовательностей без учета этого элемента и сконкатенировать полученную НОП с ним;

- Если две последовательности  $X$  и  $Y$  не заканчиваются одним и тем же символом. Тогда НОП  $X$  и  $Y$  это более длинная из последовательностей  $НОП(X_n, Y_{m-1})$  и  $НОП(X_{n-1}, Y_m)$ .

## III. Модуль контроля версий для DALL-функций в СУИ

Работа модуля мониторинга с DALL-библиотеками осуществляется следующим образом.

Необходимый для работы модуля мониторинга инструментарий находится в папке, расположение которой конфигурируется администратором системы. При запуске модуля мониторинга, либо по запросу администратора происходит синхронизация DALL-библиотек модуля мониторинга с удаленным репозиторием. Если модуль оперирует новой библиотекой, то файл записывается в базу данных целиком. Если же модуль оперирует более новой версией уже существующей библиотеки, то в базу данных записываются отличия от предыдущей версии согласно описанному выше алгоритму. Важно отметить, что алгоритм реализован в серверной части СУИ, что позволяет экономить вычислительные ресурсы устройств, на которых установлен модуль мониторинга. Также модуль контроля версий позволяет отсылать запрос на загрузку новой версии библиотеки или откат до более старой версии администратором системы. В таком случае агент шлет запрос на сервер, который, обрабатывая записи в базе данных о разнице между двумя версиями, возвращает на агент требуемую версию файла.

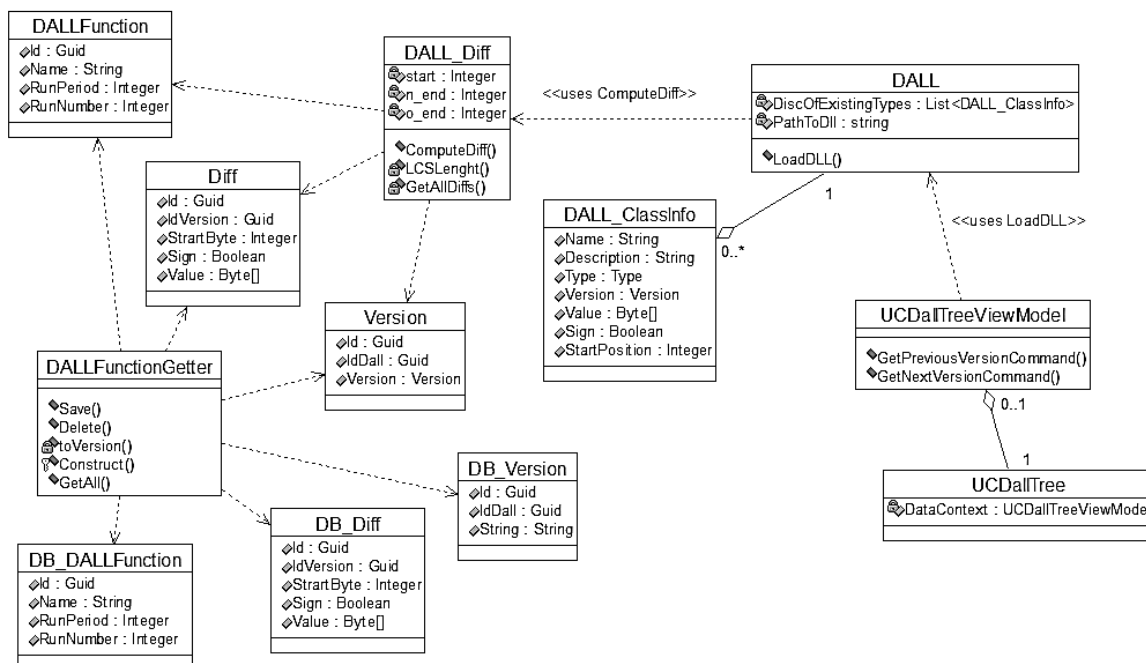


Рисунок 1 – Упрощенная структура разработанного программного модуля

Для предложенного модуля был использован шаблон программирования Model-View-ViewModel (MVVM). Класс DALL (Model) получает данные из класса, в котором имплементирован алгоритм поиска НОП (Dall\_Diff). Этот класс в свою очередь оперирует с классами, в которые отображаются сущности, хранимые в базе данных. Класс UCdallTreeViewModel (ViewModel) представляет класс DALL в виде, пригодном для отображения на пользовательском интерфейсе, а так

же содержит реализацию функционала, предоставленного администратору системы. Класс UCDAIITree являет собою визуальную часть предложенного модуля.

### Выводы

Разработан программный модуль контроля версий динамически подсоединяемых библиотек для СУИ SmartBase ITS Control. Алгоритм, который базируется на методах динамического программирования, имплементирован на удаленном сервере, отведенном под репозиторий динамически подсоединяемых библиотек, что позволяет экономить вычислительные ресурсы устройств, на которых установлен модуль мониторинга.

### Список использованных источников

1. Ролик А.И. Система управления корпоративной информационно-телекоммуникационной инфраструктурой на основе агентского подхода / А.И. Ролик, А.В. Волошин, Д.А. Галушко, П.Ф. Можаровский, А.А. Покотило // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка. — К.: «ВЕК+», 2011.
2. Maier D. The complexity of some problems of subsequences and supersequences. ACM Press. – 1978. – с. 322-366

УДК 004.65

## ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ДВОХФАЗНОГО АЛГОРИТМУ ПРОЕКТУВАННЯ МУЛЬТИБАЗОВИХ СХОВИЩ ДАНИХ

Яцишин А.Ю.

*Національний технічний університет України «Київський політехнічний інститут»*

### І. Постановка проблеми

У роботах [1] та [2] розглядається задача проектування і оптимізації мультибазових сховищ даних з врахуванням структурованості даних. Метою проектування є розміщення даних у сховищі (реляційна, багатовимірна бази даних, а також XML та NoSQL ) у відповідності з їх структурованістю, метою оптимізації є застосування методик (перерозташування структурованих та слабо структурованих даних, індексація, матеріалізація, горизонтальна та вертикальна фрагментація, злиття). Для оптимізації використовується генетичний алгоритм, у даній доповіді розглядається застосування так званої адаптивності за генами.

### II. Мета роботи

Метою дослідження є розробка двофазного алгоритму проектування мультибазових сховищ даних. У даній роботі пропонуються експериментальні результати проведеного дослідження.

### III. Дослідження двофазного алгоритму

Запишемо задачу оптимізації мультибазових сховищ даних.

Нехай  $K_1, \dots, K_n$ - класи запитів до сховища даних, що надійшли до сховища даних за звітний період,  $m(S, i)$  - умова застосування компоненту штрафу по оновленню,  $f_i^s$  - доля всіх запитів вибору даних класу  $K_i$  по відношенню до всіх запитів вибору у сховищі за звітний період,  $n_i$ - розмір даних для еталонних запитів, визначається як кількість даних, отриманих у результаті виконання запитів, або, якщо запит не повертає даних, сумарну загальну площу таблиць, що беруть участь у запиті;  $t_i^s$ - час виконання еталонного запиту вибірки даних.

Необхідно оптимізувати сховище даних, тобто знайти такий стан сховища S, що наступна функція приймає максимальне значення

$$\sum_{i=1}^n \frac{n_i f_i^s}{t_i^s(S)} \rightarrow \max, \quad (1)$$

Дана оптимізація проводиться у два етапи. На першому етапі дані розміщуються у сховищі відповідно до їх структурованості, на другому – отримане сховище оптимізується відповідно до статистики виконання запитів.

У таблиці 1 показана швидкодія запитів (1) до інтеграції даних, після неї, та після оптимізації сховища.