

М.Ф. Єфремов, к.т.н., доц.

В.М. Єфремов, здобувач

Ю.М. Єфремов, к.т.н., доц.

*Житомирський державний технологічний університет***СКЛАДАЛЬНЕ ПРОГРАМУВАННЯ ДЛЯ ПОБУДОВИ СИСТЕМ УПРАВЛІННЯ ПІДПРИЄМСТВАМИ З БАЗОВИМИ ПРОГРАМНИМИ КОМПОНЕНТАМИ**

Розглянуто головні принципи створення програм для автотранспортних підприємств з використанням готових компонентів і програмних інтелектуальних агентів на основі складального об'єктного програмування. Запропонована модель представлена у вигляді готової програми, що складається з програмних компонентів, програмних агентів і інтуїтивно зрозумілого інтерфейсу, що дозволяє кінцевому користувачеві автотранспортного підприємства перенастроювати програму відповідно до поточних вимог діючого законодавства і наказів підприємства, без подальшого залучення програмістів для написання програмного коду. В основу написання програми покладено принципи складального програмування з використанням програмних агентів, які, відстежуючи повторюваність дій користувача, пропонують автоматизувати дані дії. Також, у разі порушення повторюваності, попереджають про можливу помилку. Кожен з компонентів є повністю самостійним програмним продуктом, що може працювати як самостійно, так і у взаємодії з іншими компонентами, що дозволяє використовувати лише необхідні компоненти. Це значно прискорює виконання розрахунків по підприємству, а також не вимагає наявності дорогого обладнання для проведення цих розрахунків.

Ключові слова: ресурс; об'єкт; програмні компоненти; складальне програмування; програмні агенти; готова компонента.

Постановка проблеми. Розробка та впровадження складних програмних комплексів для автотранспортних підприємств вимагає залучення значних ресурсів як розробника, так і самого підприємства, тому вартість таких рішень для промислового підприємства середнього розміру починається від декількох десятків тисяч доларів в нижньому ціновому діапазоні і практично не має верхньої цінової межі. Виникає питання: чи є шляхи її зменшення з боку розробників системи.

Формулювання проблеми. При створенні сучасного програмного продукту програмісту не доводиться створювати всі програмні компоненти від початку до кінця, а доцільніше використовувати вже наявні у нього бібліотеки з готовими компонентами і об'єктами [1–5]. Під бібліотеками знань розуміються інтелектуальні бази даних.

Аналіз досліджень та публікацій. Наявність достатньої кількості наукових досліджень свідчить про зацікавленість науковців питанням побудови систем управління автотранспортними підприємствами. Проблему побудови даних систем досліджували: Липаєв В.В., Філінов Е.Н. [5], Маміконов А.Г. [6], Буч Г. [7], Zubesk J. [8], Трофімов С.А. [9, 12], Брукс Ф. [10], Бюрер К. [11], що відображено в опублікованих матеріалах. Проте проведені дослідження не дають відповіді на всі питання та не вирішують комплекс проблем, пов'язаних з вартістю кінцевого продукту, шляхів зниження собівартості програмних комплексів. Звідси виникає необхідність подальшого дослідження проблем, пов'язаних зі створення програм для автотранспортних підприємств з використанням готових компонентів і програмних агентів на основі складального програмування.

Виділення невирішених раніше частин загальної проблеми. Для ефективного і вмілого їх використання необхідно чітко розуміння принципу функціонування та управління кожного з використовуваних об'єктів або компонент, щоб на основі такої структури значно зменшити трудомісткість створення і, що важливо, супроводу системи, вберегти від дорогих логічних помилок при створенні програмних модулів.

Мета досліджень. Використання складального програмування для побудови систем управління підприємствами з базовими програмними компонентами з використанням інтелектуальних агентів.

Викладення основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів. Для цього необхідно, щоб бібліотека готових компонентів і об'єктів була представлена у вигляді двовірної матриці, де поля такої матриці були б посиланнями на готові методи і об'єкти. Вони повинні бути впорядковані за певними сукупним ознаками:

$$LKL = \begin{vmatrix} a[L1,C1] & a[L1,C2] & a[L1,Cn] \\ a[L2,C1] & a[L2,C2] & a[L2,C(n+1)] \\ a[Lm,C1] & a[L(m+1),C2] & a[L(m+x),C(n+y)] \end{vmatrix},$$

де LKL – Link Know Lib (Пов'язана бібліотека знань);

Li, Cj – впорядкування координати на посилання методу;

за Li – класу; Cj – за групою;

A[Li, Cj] – покажчик на місце знаходження методу.

Таким чином, кожен з методів даної матриці буде являти собою абсолютно автономну структуру, незалежну від сусідніх полів матриці і мати лише власні методи управління та обміну інформацією. Ця властивість дозволяє зберігати дану бібліотеку у вигляді динамічно пов'язуваної бібліотеки стандарту (DLL – dynamic link lib), що, в свою чергу, надає можливість оновлювати її та доповнювати новими елементами знань практично на будь-якій стадії експлуатації.

Готова компонента (ГК) представляється у вигляді завершеного процесу, що реалізує певні дії, а також володіє власними методами управління і здійснює обмін даними по інформаційному каналу зв'язку. Усередині готової компоненти деякі методи управління можуть бути реалізовані у вигляді абстрактних і віртуальних методів. Ця властивість дозволяє адаптувати ці методи в нестандартних випадках використання, що розширить можливості до її експлуатації. Схематично взаємодія між готовими компонентами в готовій програмі представлена на рисунку 1.

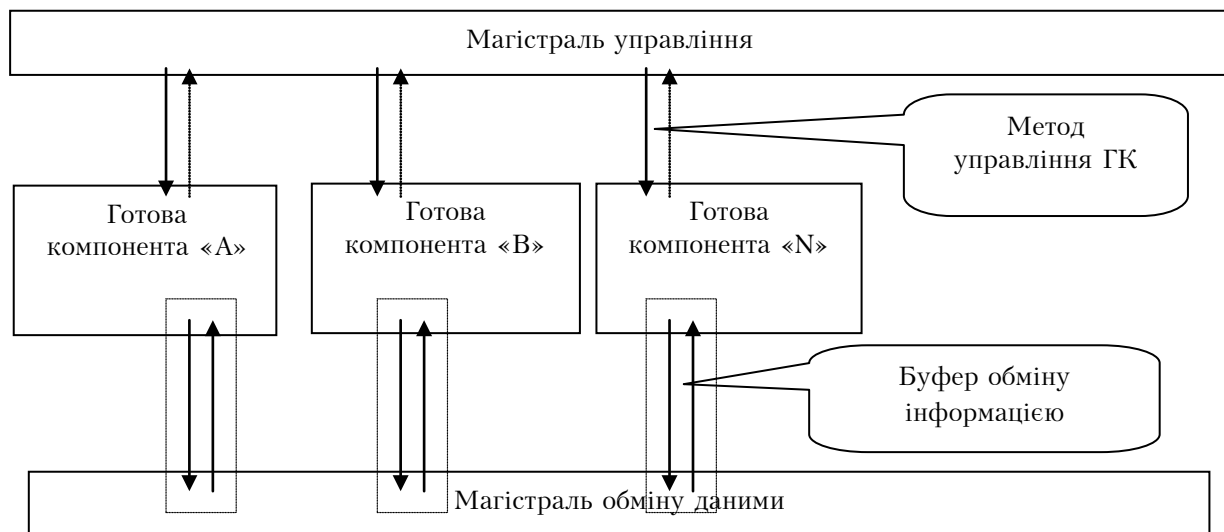


Рис. 1. Схема внутрішньої взаємодії між готовими компонентами в програмі

Таким чином, згідно з рисунком 1, готові компоненти не обмінюються інформацією безпосередньо між собою, що забезпечує цілісність отримання даних від компоненти до компоненті за рахунок магістралі даних, до якої повинен бути визначений конкретний протокол обміну.

Процедура взаємодії між компонентами виглядає наступним чином: магістраль керування готовою програмою формує запит на активування конкретної готової компоненти, з метою отримання конкретних даних, готової компоненти аналізує мета запиту і повертає список необхідних ще готових компонент для повного формування відповіді на вхідний запит. Дана стадія підготовки має назву ініціалізації готової компоненти. Потім магістраль керування активізує всі необхідні допоміжні готової компоненти для можливості обміну інформацією через магістраль обміну інформацією. Дана стадія має назву активація системи. Кількість

допоміжних готових компонент має бути визначена на стадії розробки програми. І, нарешті, на останній стадії виконання відбувається остаточне формування всіх необхідних запитуваних даних.

Даний вид обміну інформацією дозволяє проектувати програми з розподіленими базами даних та забезпечує підтримку як у локальних, так глобальних інформаційних мережах. Програмісту, який проектує завдання відповідно до технологій складального програмування, необхідно лише правильно визначити протокол, за яким буде відбуватися обмін даними (зі стандартизованих найбільш розповсюдженими та документованими є протоколи IPX/SPX або TCP/IP тощо) або розробити за необхідності власний.

Таким чином, надання програмісту готової структури ПЗ, побудованої на основі об'єктивних досліджень конкретної предметної області, і методичних рекомендацій зі створення ПЗ на основі такої структури, може значно зменшити трудомісткість створення і, що важливо, супроводу системи, вберегти від дорогих логічних помилок при створенні програмних модулів. Така структура була розроблена, а методика створення ПЗ на основі модулів, побудованих за даній структурі – базових програмних компонент (БПК) (рис. 2) – була впроваджена в Житомирській службі автомобільних доріг.

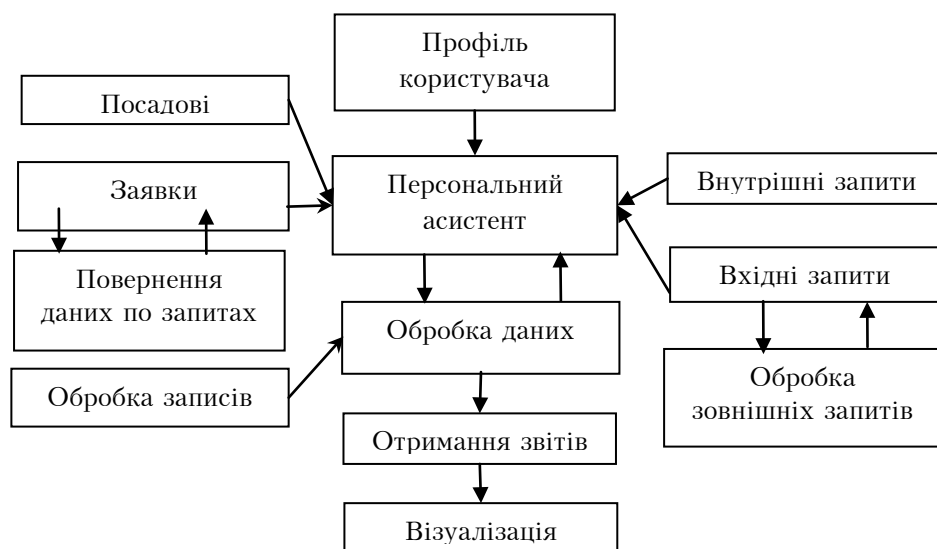


Рис. 2. Функціональна схема БПК

Термін "базовий програмний компонент" визначається як набір програмних об'єктів, що враховують особливості предметної області, є невіддільними один від одного в межах виконання певного класу завдань, мають уніфікований інтерфейс, що дозволяє БПК взаємодіяти між собою без додаткового координуючого коду та мають можливість зміни своїх властивостей без внесення змін в програмний код. Основою БПК при такому підході є динамічно створюване уявлення даних, що побудоване на роботі з агентами. Кожен користувач, який працює з системою, має свій пароль доступу і одночасно отримує програмного агента або, як його ще називають, персонального асистента (ПА), який запам'ятовує послідовність роботи користувача, записуючи його у своїх властивостях, використовуючи для цього збережену в профілі користувача інформацію. Ці метадані є описувачами оброблюваних полів таблиць, шаблонів введення-виведення і посилань на інші таблиці. При цьому всі алгоритми виводу на друк, пошуку, угруповання, додавання, видалення записів реалізовані незалежно від оброблюваної інформації. Для заповнення посилальної частини таблиць передбачений інтерфейс для взаємодії з тим самим модулем, але налаштованим для роботи з іншими даними, тобто передбачається рекурсивний виклик.

Для створення ПЗ на основі БПК пропонуються наступні кроки:

1. Створення БПК – найбільш трудомісткий і дорогий етап, який повинен бути виконаний кваліфікованим програмістом. Він містить наступні пункти:

- • вибір засобів створення (RAD, CASE);
- • визначення необхідного набору функцій;

- • визначення необхідного обсягу адаптації;
- • створення ієрархії класів;
- • створення виконуваного коду БПК.

2. Аналіз предметної області для подальшого створення модулів системи на основі БПК. Цей етап виконується аналітиком системи і не містить безпосереднього програмування. Його можна розбити на наступні пункти

- визначення набору інформаційних журналів для роботи БПК;
- визначення структури інформаційних журналів, форматів полів даних і зв'язків;
- визначення функцій і кваліфікації користувачів.

3. Модулі системи на основі БПК і профілів користувачів. Цей етап виконується програмістами середньої кваліфікації й містить основний обсяг робіт зі створення системи. Цей етап розбивається на наступні пункти:

- створення структури для динамічного формування уявлень даних і за необхідності програмних модулів для їх обробки на основі шаблонів БПК;
- заповнення структури для динамічного формування уявлень даних згідно зі структурами інформаційних журналів, форматами полів і зв'язків;
- попереднє заповнення профілів користувачів відповідно до функцій і кваліфікації користувачів.

4. Налаштування ПЗ здійснюється відділом впровадження та, надалі, відділом супроводу безпосередньо на підприємстві і містить наступні пункти:

- визначення конфігурації дискових пристроїв для конкретних робочих станцій;
- заповнення еталонів адрес зберігання файлів;
- остаточна настройка профілів користувачів відповідно до функцій та кваліфікації конкретних користувачів з налаштуванням форм вводу–виводу.

Висновки та перспективи подальших досліджень в даному напрямку. Запропонована методика дозволяє скоротити витрати на створення і супровід ПЗ АСУП для автотранспортних підприємств:

- шляхом скорочення розмірів програмного коду, що вимагає подальшого супроводу;
- підвищення його гнучкості, що дозволяє здійснювати основну адаптацію до мінливих умов експлуатації силами самого підприємства без залучення розробника;
- зниження вимог до кваліфікації програмістів;
- підвищення надійності системи за рахунок використання багаторазово протестованих компонентів.

Список використаної літератури:

1. Єфремов В.М. Побудова систем управління автогосподарством із застосуванням підходу багатоагентного програмування / В.М. Єфремов // Вісник ЖІТІ. – 2000. – № 12 / Технічні науки. – С. 234–237.
2. Єфремов В.М. Предметна область та основні завдання автоматизації автогосподарства / В.М. Єфремов // Вісник ЖДТУ. – 2008. – № 3 (46) / Технічні науки. – С. 113–120.
3. Єфремов М.Ф. Комплексна система інтелектуалізації управлінської праці в автогосподарствах на базі інтелектуальних агентів / М.Ф. Єфремов, В.М. Єфремов, Ю.М. Єфремов // Вісник ЖДТУ. – 2008. – № 4 (47) / Технічні науки. – С. 139–146.
4. Єфремов М.Ф. Автоматизація управління автогосподарством на базі агентів з врахуванням міжнародних перевезень / М.Ф. Єфремов, В.М. Єфремов, Ю.М. Єфремов // Вісник ЖДТУ. – 2010. – № 1 (47) / Технічні науки. – С. 80–85.
5. Липаев В.В. Мобильность программ и данных в открытых информационных системах / В.В. Липаев, Е.Н. Филинов. – М. : Научная книга, 1997. – 368 с.
6. Мамиконов А.Г. Методы разработки автоматизированных систем управления / А.Г. Мамиконов. – М. : Энергия, 1973. – 336 с.
7. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г.Буч : пер. с англ. – М.–СПб. : Издат. Бином ; Невский диалект, 1999. – 560 с.
8. Zubeck J. Повторное использование объектов в системах укоренной разработки приложений / J.Zubeck // Computerweekly. – 1998. – № 7. – С. 24–28.

9. Трофимов С.А. CASE-технологии: практическая работа в Rational Rose / С.А. Трофимов. – М. : ЗАО Изд-во БИНОМ, 2001. – 272 с.
10. Брукс Ф. Мифический человек-месяц или как создаются программные системы / Ф.Брукс : пер. с англ. – СПб. : Символ-Плюс, 1999. – 304 с.
11. Бюрер К. От ремесла к науке: поиск основных принципов разработки ПО / К. Бюрер [Электронный ресурс]. – Режим доступа : <http://interface.ru/rational/science.htm>.
12. Трофимов С. Методика создания программного обеспечения для систем управления предприятиями с использованием типовых программных компонентов / С. Трофимов [Электронный ресурс]. – Режим доступа : <http://www.caseclub.ru/articles/metodpo.html>.

ЄФРЕМОВ Микола Федорович – кандидат технічних наук, доцент кафедри ПЗС Житомирського державного технологічного університету.

Наукові інтереси:

- програмування;
- штучний інтелект.

Тел.: 0936475220.

E-mail: eyuri@list.ru.

ЄФРЕМОВ Віталій Миколайович – здобувач Житомирського державного технологічного університету.

Наукові інтереси:

- програмування;
- штучний інтелект.

Тел.: +380936475221

E-mail: eyuri@list.ru

ЄФРЕМОВ Юрій Миколайович – кандидат технічних наук, доцент кафедри ПЗС Житомирського державного технологічного університету.

Наукові інтереси:

- програмування;
- штучний інтелект.

Тел.: +380936475222.

E-mail: eyuri@list.ru.

Стаття надійшла до редакції 12.10.2015.