

МЕТОД ВИБОРУ ШАБЛОНІВ ПРОЕКТУВАННЯ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМ СТРУКТУРНОЇ ОРГАНІЗАЦІЇ КОДУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

У статті представлено метод визначення шаблону проектування вихідного коду програмного забезпечення. Базуючись на запропонованому підході розроблено методологічні принципи, модель та алгоритм проектування програмних систем, та створення на цій основі адаптивного середовища пошуку рішень в умовах необхідності постійної модернізації та реструктуризації елементів програмних комплексів, систем і т.д.

Запропоновано метод та розроблено алгоритм визначення шаблону проектування вихідного коду програмного забезпечення, який дозволить обирати найбільш підходящий шаблон проектування, що на ранніх етапах розробки програмного забезпечення вирішуватиме проблему структуризації вихідного коду

Ключові слова: шаблон, програмний код, проектування.

Вступ. На сучасному етапі розвитку ринкової економіки актуальними проблемами залишаються проблеми створення нового програмного забезпечення та комп'ютерних систем а також, не менш важливими проблемами, залишається модернізація та реструктуризація вже створених та працюючих програм та систем [1,2]. Швидкі зміни зовнішніх умов потребують удосконалення технологій, розширення функціоналу, збільшення швидкодії і т.д. Це призводить до необхідності структурно-параметричної перебудови програмної системи у відповідності з поставленими цілями, що пов'язано з вирішенням узгоджених завдань проектування на всіх етапах розробки. При цьому основою для формування нових проектних варіантів являється аналіз вже діючих систем.

Складність формалізації та рішення задач проектування програмних систем обумовлена, на сам перед, необхідністю моделювання та тестування великої кількості взаємопов'язаних елементів системи, комплексного обліку обмежень на ресурси та технологію їх розвитку. В основі пошуку раціональних методів проектування програм та систем лежить комплекс рішень взаємопов'язаних оптимізаційних рішень у відповідності з поставленими цілями та обмеженнями. Також можна виділити задачі структурного синтезу які передбачають склад структурних елементів та зав'язків між ними, і задачі параметричного синтезу які орієнтовані на вибір оптимальних параметрів системи при її фіксованій структурі [3].

Визначення проблеми. При проектування програм та програмних систем, в якості основних структурних елементів розглядають модулі обробки даних, модулі користувацького інтерфейсу, модулі збереження даних і т.д.. Узагальнена структура може бути представлена як результат об'єднання модулів обробки даних та модулів збереження даних через модулі користувацького інтерфейсу [2,3].

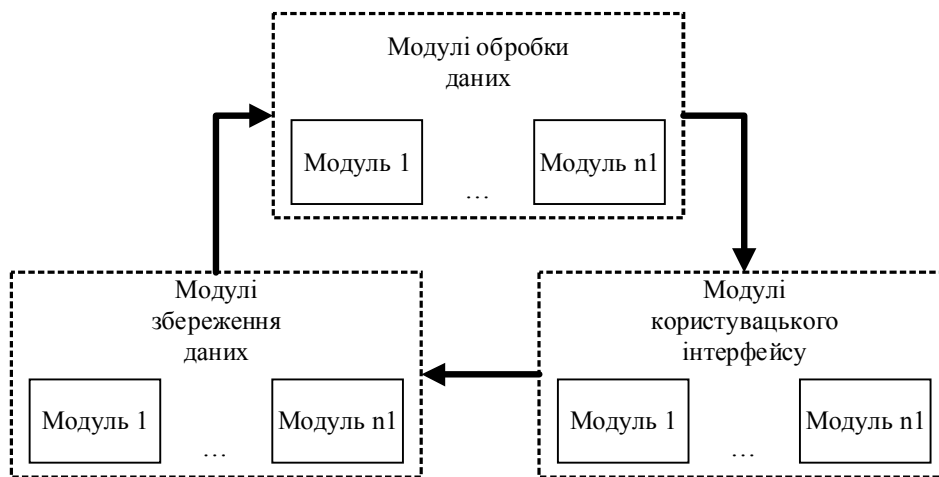


Рис. 1. Узагальнена структура програмної системи

Структура програмної системи залежить від форм та методів організація процесів пов'язаних з обробкою та збереження даних, від кількості даних, їх спеціалізації, кооперації та комбінування. Глибина розділення процесу обробки даних на його складові частини, та ступінь кооперації визначають види та призначення компонентів програми.

Першим етапом розробки нової програмної системи або удосконалення вже існуючої, є комплексний всебічний аналіз всіх характеристик яким система повинна відповідати. Для того щоб сформувавши загальне уявлення в цілому про програмну систему як про систему яка проектується, необхідно проаналізувати наступні характеристики [1]:

Група 1:

- Алгоритмічна складність (логіка алгоритмів обробки інформації);
- Склад та глибина роботи функцій обробки які будуть реалізовані;
- Повнота та системність функцій обробки;
- Об'єм файлів програми;
- Вимоги до операційної системи та до технічних засобів обробки програмними засобами;
- Об'єм дискового простору;
- Необхідна кількість оперативної пам'яті для запуску;
- Тип процесора;
- Версія операційної системи;
- Наявність обчислювальної мережі;

Наступна ступінь диференціації характеристик виникає при переході до аналізу окремих компонентів програми. При цьому відбувається деталізація характеристик першої групи. Так, структура програми деталізується у вигляді схем основних технологічних взаємозв'язків в середині кожного з модулів.

До другої групи характеристик відносяться характеристики форм організації основних функціональних процесів програми:

Група 2:

- Технології, мови програмування за допомогою яких буде проводитись розробка;
- Організація зв'язків між компонентами програми;
- Створення інтерфейсів для комунікації між компонентами створених з використанням різних технологій або мов програмування;
- Специфікація даних з якими буде працювати той чи інший компонент системи.
- Рівень завантаженості каналів зв'язку;
- Характер передачі даних між компонентами програми;
- Спосіб зберігання даних;
- Швидкість обробки даних;

На основі аналізу цих характеристик визначаються напрямки створення програмної системи та визначаються можливі напрямки розвитку при її подальшій модернізації.

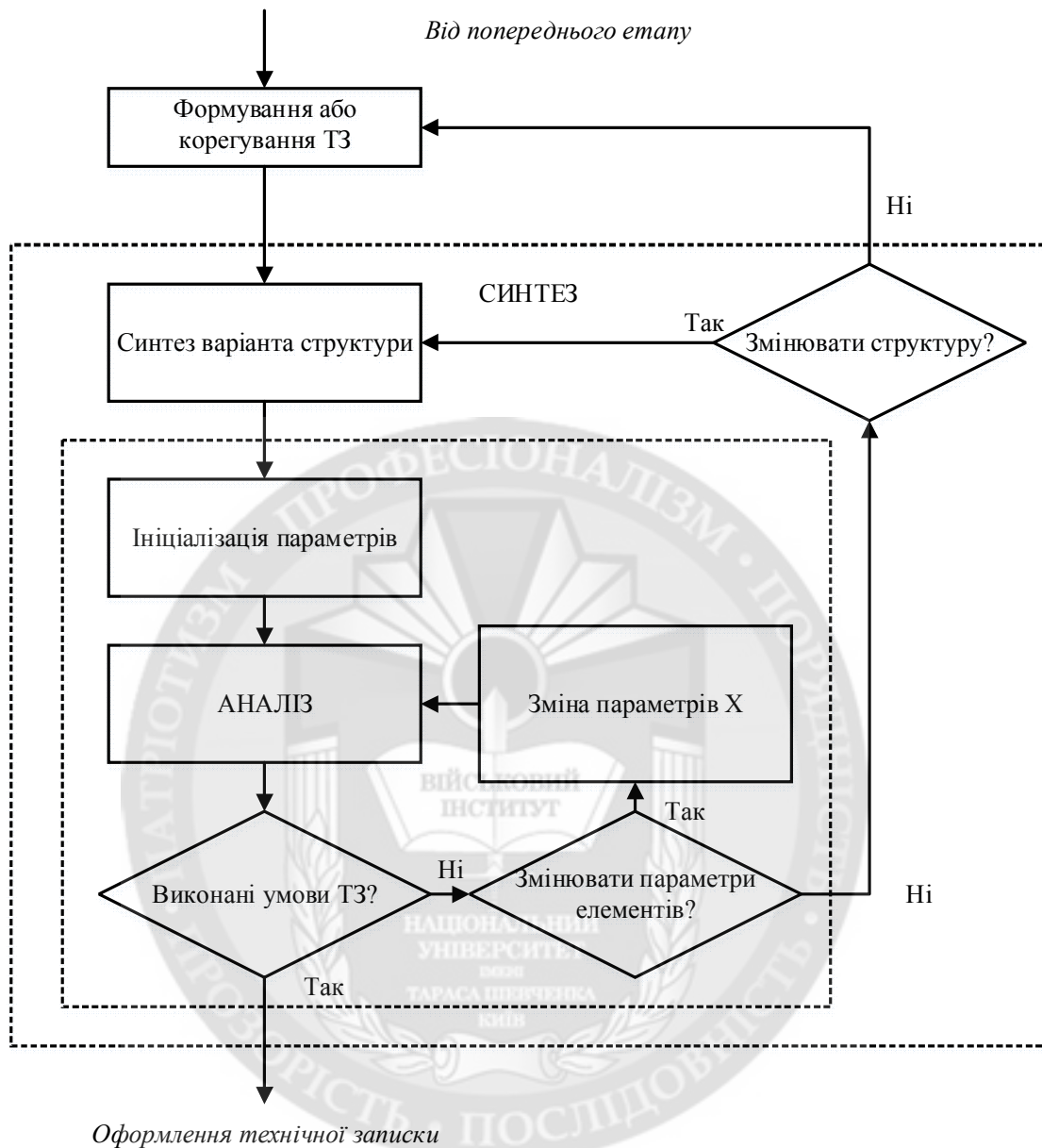


Рис. 2. Схема типового маршруту проектування

На рис. 2 показана схема процесу проектування програмної системи яка відповідає трьом рівням оптимізації. Перший рівень полягає у виборі найкращої ідеї або принципу дії об'єкта проектування; другий рівень – у пошуці найкращого структурного рішення або принципу дії; третій рівень – у визначенні найкращих параметрів для вибраної структури. Задачі першого рівня характерні для етапу зовнішнього проектування і вирішуються з використання експертних підходів і методів. Математичне забезпечення сучасних САПР орієнтовано на рішення задач другого та третього рівня. Отримання оптимального рішення пов'язано з вибором найкращого варіанта з деякої допустимої множини рішень. Склад оптимізаційного процесу являє собою набір варіантів, які відповідають вимогам до функціонування програмної системи, а також набору експлуатаційних і технічних обмежень.

Формалізація задач оптимального проектування програмних систем та програм в загальному випадку зводиться до їх формулювання у вигляді наступної оптимізаційної моделі [2,3]:

$$f_i(X) \rightarrow \min_{X \in D}, i = \overline{1, m}$$

$$D = \{X / x_j^{\min} \leq x_j \leq x_j^{\max}, j = \overline{1, n}; g_p(X) \geq 0, h_k(X) = 0, \\ p = \overline{1, s}, k = \overline{1, q}\}$$

Тут $X = (x_1 \dots x_n)$ – вектор змінних, які визначають структуру та параметри системи; $f_i(X)$ – критерії оптимальності; D – область допустимих обмежень двох видів: прямим $x_j^{\min} \leq x_j \leq x_j^{\max}$ та функціональним $g_p(X) \geq 0, h_k(X) = 0$. Обмеження визначають технічно-економічні вимоги до системи що проектується.

Особливістю програмних систем являється неможливість їх адекватного опису за допомогою аналітичних оптимізаційних моделей. Сучасні програмні системи відносяться до класу складних систем і характеризуються множиною взаємозв'язків між елементами, неоднозначністю алгоритмів поведінки при різних умовах, наявністю безлічі випадкових впливів, кореляційної взаємодії між великою кількістю параметрів [4]. Все це ускладнює аналітичне формулювання критеріїв оптимальності та обмежень в оптимізаційних моделях. Моделі що описують залежності між вхідними, вихідними та внутрішніми параметрами в аналітичній формі використовуються лише в рідкісних випадках проектування окремих елементів програмної системи при наявності ряду полегшуючих припущень.

На даний час комплексний обрахунок особливостей складних програмних систем в процесі їх проектування забезпечується на основі використання оптимізаційно-імітаційного підходу до проектування. Цей підхід заснований на використанні оптимізаційних процедур в поєднанні з імітаційними моделями при рішенні задач оптимального проектування. Його використання дозволяє проектувати програмні системи з урахуванням динамічних і стохастичних аспектів їх функціонування і приводить до специфічних ітеративних схем пошуку оптимальних варіантів (рис. 3).

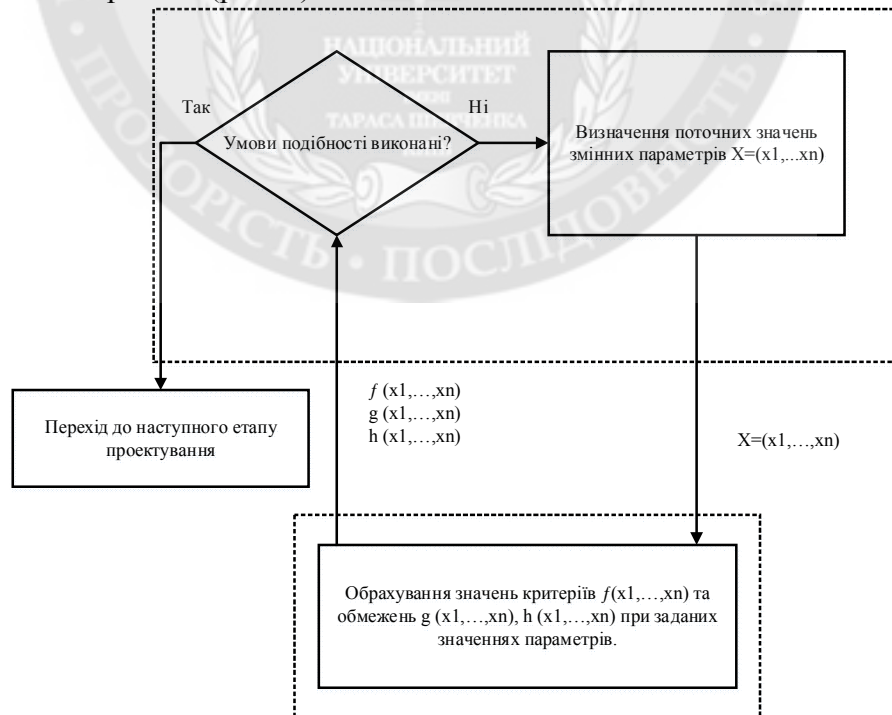


Рис. 3. Взаємозв'язок процедур імітаційного моделювання та оптимізації під час пошуку раціональних проектних рішень

При реалізації даного підходу оптимізаційні процедури використовуються для синтезу чергового варіанту параметрів або структури проекрованої системи, а блок імітаційного моделювання – для визначення вихідних характеристик системи при заданих значеннях параметрів. Вихідні характеристики використовуються для формування критеріїв і обмежень в оптимізаційній моделі. Процес пошуку оптимальних проектних рішень при цьому заснований на вирішенні послідовності задач моделювання та аналізу при різних умовах. З урахуванням результатів аналізу виконується ітераційний процес покращення першочергового проектного рішення на основі зміни значень параметрів оптимізаційної моделі [5,6].

Використання оптимізаційно-імітаційного підходу при проектуванні програмних систем передбачає вирішення наступних основних завдань [7]:

- Вибір технології імітаційного моделювання, відповідних математичних схем, інструментальних засобів і реалізації імітаційної моделі програмної системи у вибраному середовищі з можливістю її оперативної модернізації.

- Розробка алгоритмів оптимального проектування програмних систем, що дозволяють проводити пошук проектних рішень при відсутності аналітичних формулювань критеріїв оптимальності та обмежень в оптимізаційних моделях.

- Розробка технологій і схем інтеграції та узгодженості оптимізаційних процедур та імітаційних моделей в процесі проектування.

Створення на основі розроблених методів та алгоритмів математичного та програмного забезпечення підтримки прийняття рішень, дозволить підвищити ефективність процесу оптимального проектування складних програмних систем.

Подальші дослідження будуть спрямовані на практичну реалізацію методу підбору проектних рішень розробки програмних систем.

Висновок: Розробка нових методів проектування які засновуються на оптимізаційному підході проектування дозволить полегшити процес проектування та модернізації вже існуючих програмних систем. Використання методу підбору шаблонів проектування дозволить скоротити час планування процесу розробки програмного забезпечення та значно полегшити його розробку.

ЛІТЕРАТУРА:

1. Багриновский К.А. Современные методы управления технологическим развитием / К. Багриновский, М. Бендинов, Е. Хрусталеv. – К: РОССПЭН, 2001. – 272 с.
2. Батищев Д.И. Оптимизация в САПР /Батищев Д, Я. Львович, В. Фролов. – К.: Воронеж: Издательство ВГТУ, 1997. – 416 с.
3. Белецкая С.Ю. Анализ и синтез стохастических технологических систем / С. Белецкая, Н. Боковая, Д. Десятов – К: Воронеж: Научная книга, 2010. – 236 с.
4. Воронин А.Н. Нелинейная схема компромиссов в многокритериальных задачах оценивания и оптимизации / Воронин А.Н. – К: Кибернетика и системный анализ, 2009. – Т. 45. – №2. – С. 106-115
5. Норенков И.П. Основы автоматизированного проектирования / Норенков И.П. – К: МГТУ им. Баумана, 2002. – 336 с.
6. Советов Б.Я. Моделирование систем / Б. Советов, С. Яковлев. – К: Высш. Шк., 2001. – 343 с.
7. Хоботов Е.Н. Использование оптимизационно-имитационного подхода для моделирования и проектирования производственных систем / Хоботов Е.Н. – К: Автоматика и телемеханика, 1999. – №8. – С. 163-176.

Рецензент: д.т.н., проф. Мясіщев О.А., професор кафедри комп'ютерних систем та мереж, Хмельницький національний університет

Гнатюк В.В., Муляр И.В., Гурман И.В.

**МЕТОД ВЫБОРА ШАБЛОНОВ ПРОЕКТИРОВАНИЯ ДЛЯ РЕШЕНИЯ ПРОБЛЕМ
СТРУКТУРНОЙ ОРГАНИЗАЦИИ КОДА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
КОМПЬЮТЕРНЫХ СИСТЕМ**

В статье представлен метод определения шаблона проектирования исходного кода программного обеспечения. Основываясь на предложенном подходе разработаны методологические принципы, модель и алгоритм проектирования программных систем, и создание на этой основе адаптивной среды поиска решений в условиях необходимости постоянной модернизации и реструктуризации элементов программных комплексов, систем и т.д.

Предложен метод и разработан алгоритм определения шаблона проектирования исходного кода программного обеспечения, который позволит выбирать наиболее подходящий шаблон проектирования, что на ранних этапах разработки программного обеспечения решать проблему структуризации исходного кода.

Ключевые слова: шаблон, программный код, проектирование.

Ph.D. Hnatiuk V.V., Ph.D. Mulyar I.V., Gurman I.V., Zhyrov B.G.

**THE METHOD OF SELECTING DESIGN PATTERNS TO ADDRESS THE STRUCTURAL
ORGANIZATION CODE SOFTWARE COMPUTER SYSTEMS**

The paper presents the method of determining the pattern design source software. Based on the proposed approach the methodological principles model and algorithm design software systems, and the establishment of the basis of adaptive environment in terms of finding solutions need constant modernization and restructuring items of software systems, systems, etc .

The method and algorithm definition pattern design source software that allow to choose the most suitable design pattern that in the early stages of software development will solve the problem of structuring source code

Keywords: template code, design.