

УДК 621.391:681.5

Б. Ю. ЖУРАКОВСЬКИЙ, д-р техн. наук, професор,
Державний університет телекомунікацій, Київ

ПОБУДОВА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ СИСТЕМИ УПРАВЛІННЯ ІНФОКОМУНІКАЦІЙНОЮ МЕРЕЖЕЮ

Досліджено актуальні проблеми визначення об'єкта управління, що набуває особливої ваги при проектуванні та побудові системи управління. Розглянуто основні складові частини об'єктно-орієнтованої методології: об'єктно-орієнтований аналіз; об'єктно-орієнтоване проектування; об'єктно-орієнтоване програмування. Визначено такі поняття, як діаграма класів, статична і динамічна модель об'єктно-орієнтованої програмної системи. Наведено варіанти графічного зображення класу на діаграмі класів. Запропоновано контекстну діаграму класів для розв'язування комбінаторно-оптимізаційних задач у системі управління інфокомунікаційною мережею.

Ключові слова: об'єктно-орієнтована модель; системи управління мережею; об'єктно-орієнтований аналіз; моделі станів; діаграма класів.

Постановка задачі

Останніми роками розробникам доводиться дедалі частіше розв'язувати задачі проектування складних систем управління об'єктами інфокомунікацій.

Тому об'єкти, для яких постає завдання з вироблення нових концепцій при побудові систем управління (СУ), тобто великих систем, становлять особливий інтерес. Проаналізуємо характерні риси СУ та відповідних моделей, які виникають при проектуванні таких систем.

1. Здатність еволюціонувати в часі. Структура таких систем і їхнє функціонування зазнають тих чи інших змін у часі. Нині телекомунікаційна мережа, яка виникла внаслідок поступового перетворення розрізнених локальних мереж на єдину інформаційну інфраструктуру, використовується не лише за своїм прямим призначенням, але і як мережа передавання даних між обчислювальними машинами.

2. Наявність великої кількості суперечливих критеріїв управління. Як правило, ці критерії навіть не підлягають чіткому формулюванню. Це, насамперед, *завадостійкість управляючої інформації, швидкість, затримка, вартість.*

3. Присутність об'єктів управління (ОУ), поведінка яких істотно визначає поведінку всього об'єкта. На відміну від автоматів, ОУ мають не жорстко детерміновану, а пов'язану з багатьма можливими режимами роботи поведінку. Це ускладнює прогнозування поведінки ОУ, особливо тоді, коли є великий спектр зовнішніх впливів. У цьому разі доцільно використовувати *вірогідність моделі прогнозування.*

4. При побудові моделі СУ в об'єктно-орієнтованому аналізі брали за початковий не унікальний, а деякий типовий об'єкт. Тільки-но модель об'єкта (наприклад, у вигляді діаграми станів) було побудовано, її можна було використовувати для всіх об'єктів аналогічного типу. Що ж до моделі управління первинною мережею, то її не можна використовувати при управлінні сучасною

інфокомунікаційною мережею. У кожному конкретному випадку модель має складатися заново, оскільки в ній мають бути відбиті всі особливості, притаманні саме даному об'єкту. Окрім того, з огляду на еволюціонування структури та функціонування об'єкта потрібно, аби модель була відкритого типу, що у процесі свого функціонування має збагачуватися і змінюватися.

Об'єктно-орієнтована методологія розробки систем

Об'єктно-орієнтована технологія розвивається в різних галузях обчислювальної техніки як засіб розв'язання проблем, зумовлених складністю створюваних систем. Об'єктний підхід застосовується не лише в програмуванні, а й у проектуванні інтерфейсу користувача, баз даних, баз знань і навіть комп'ютерної архітектури [1].

Сенс такого широкого підходу полягає в тому, що він дозволяє застосувати об'єктну орієнтацію для розв'язання всього кола проблем, з якими доводиться стикатися при дослідженні складних систем. В основу об'єктно-орієнтованого проектування покладено уявлення про те, що програмну систему необхідно проектувати як сукупність об'єктів, що взаємодіють один з одним, розглядаючи кожний об'єкт як примірник певного класу. При цьому класи, у свою чергу, утворюють ієрархію.

Підвищення інтересу розробників до цієї методології зумовлено тим, що методи структурного аналізу і проектування не забезпечують подальшого зниження трудомісткості розробки. Об'єктно-орієнтований підхід найбільш природно відповідає реальному процесу розробки систем, і не лише програмних. Зазначений процес є ітеративний і може потребувати внесення змін до вже розроблених і налагоджених компонентів системи [2].

Як складові частини об'єктно-орієнтованої методології (ООМ) розглядають:

- об'єктно-орієнтований аналіз;
- об'єктно-орієнтоване проектування;
- об'єктно-орієнтоване програмування.

Об'єктно-орієнтований аналіз

На об'єктний підхід вплинули попередні етапи розвитку програмних засобів. Традиційні прийоми структурного аналізу ґрунтуються на потоки даних у системі [3].

Об'єктно-орієнтований аналіз (ООА) має на меті створення моделей, ближчих до реальності, із використанням об'єктно-орієнтованого підходу; це методологія, згідно з якою вимоги формуються на основі понять класів і об'єктів, що становлять словник предметної області.

За результатами ООА формуються моделі, на яких ґрунтується об'єктно-орієнтоване проектування, котре, у свою чергу, створює основу для остаточної реалізації системи з використанням методології об'єктно-орієнтованого програмування [4].

Головні переваги ООМ порівняно зі структурними методами такі [4]:

- можливість подолати обмеження, пов'язані зі складністю розроблених систем;
- використання на стадії аналізу моделей, близьких до реальності;
- застосування при аналізі та проектуванні як інформаційних систем, так і систем реального часу й апаратно-програмних комплексів;
- забезпечення можливості повторного використання розробленого програмного забезпечення, що дозволяє істотно скорочувати терміни й знизити витрати на розробку кожної наступної системи;
- підтримання ітеративного, а не лавиноподібного процесу проектування, як у разі структурного підходу;
- природна робота з різноманітною інформацією, що використовується в мультимедійних системах;
- створення більш відкритих систем;
- повне використання описових можливостей об'єктно-орієнтованих мов програмування.

Об'єктно-орієнтоване проектування

Об'єктно-орієнтоване проектування — це методологія проектування, що поєднує в собі процес об'єктної декомпозиції з прийомами подання логічної і фізичної, а також статичної і динамічної моделей проектованої системи.

Це визначення включає в себе два важливі моменти:

- 1) об'єктно-орієнтоване проектування передбачає об'єктно-орієнтовану декомпозицію;
- 2) використовуються різноманітні прийоми подання моделей, що відбивають логічну (структури класів і об'єктів) та фізичну (архітектура модулів і процесів) структуру системи.

Саме підтримкою об'єктно-орієнтованої декомпозиції об'єктно-орієнтоване проектування відрізняється від структурного проектування.

Об'єктно-орієнтоване програмування

Об'єктно-орієнтоване програмування — це методологія програмування, яка передбачає подання програми у вигляді сукупності об'єктів, кожний з яких є реалізація певного класу, а класи утворюють ієрархію на принципах успадкування [6].

У цьому визначенні можна виокремити три тези:

- 1) об'єктно-орієнтоване програмування як елементи конструкції використовує об'єкти, а не алгоритми;
- 2) кожний об'єкт є реалізація певного класу;
- 3) класи організовано ієрархічно.

Принципи об'єктного підходу

♦ Об'єктна модель як концептуальна база об'єктно-орієнтованої методології має **чотири головні елементи**:

- 1) абстрагування;
- 2) обмеження доступу, або інкапсуляція;
- 3) модульність;
- 4) ієрархія.

Без будь-якого з цих елементів модель не буде об'єктно-орієнтованою.

♦ Поряд із головними розглядають **три додаткові елементи**:

- 1) типізація;
- 2) паралелізм;
- 3) зберігання або стійкість (збереження).

Називаючи ці елементи додатковими, наголошуємо, що вони корисні в об'єктній моделі, але не обов'язкові.

Побудова концептуальної моделі предметної області

Для побудови концептуальної моделі предметної області використовують **діаграми класів**. Концептуальні моделі оперують поняттями предметної області, атрибутами цих понять і відношеннями між ними. Поняттями в предметній області можуть виступати як матеріальні предмети, так і абстракції, що їх застосовують фахівці предметної області [6].

До основних понять у моделі відносять **класи**. Клас при цьому традиційно розуміють як сукупність загальних ознак заданої групи об'єктів предметної області. Згідно з цим визначенням на діаграмі класів кожному класу відповідає група об'єктів, загальні ознаки яких і фіксує клас. Так, клас *Студент* об'єднує загальні ознаки групи людей, які навчаються у вищих навчальних закладах. Представник класу, або об'єкт (наприклад, Іваненко І. І.) неодмінно має володіти всією сукупністю ознак свого класу і може мати власні ознаки, не зафіксовані в класі. Скажімо, окрім того, що Іваненко І. І. є студент, він ще може бути

спортсменом, музикантом тощо. Строго кажучи, такою власною ознакою, що ідентифікує студента, виступає його ім'я.

На практиці визначення основних понять предметної області, які мають бути подані на контекстній діаграмі у вигляді класів,— це зовсім не тривіальне завдання. Зазвичай діють у такий спосіб:

- формують множину понять-кандидатів із іменників, що характеризують предметну область в опису варіантів використання;

- вилучають поняття, не істотні для цього варіанта використання.

Для визначення множини понять-кандидатів корисно використовувати наведений далі перелік можливих категорій понять-кандидатів.

Категорія понять-кандидатів	Приклади
Фізичні або матеріальні об'єкти	Система управління як єдине ціле
Специфікації, елементи дизайну чи опису об'єкта — зберігаються в системі, навіть за відсутності об'єктів	Конфігурація, параметри, функції
Місцезнаходження	Інфокомунікаційна мережа
Роль людини	Отримувач послуг, адміністратор мережі, абонент
Контейнери інших об'єктів	Система управління як сукупність частин, параметри як сукупність описів
Вміст контейнерів	Частина, елемент
Інші комп'ютери або системи, зовнішні щодо системи, яка розробляється	Система транспортування інформації
Абстрактні поняття	Управляюча інформація
Організації	Фірма, підприємство
Події	Передавання управляючої інформації, передавання даних
Процеси та їхні частини (подаються у вигляді класу, якщо елементи процесу не зазнають аналізу)	Очікування, приймання повідомлення, передавання повідомлення
Правила і політика	Правила роботи системи управління
Записи фінансової, трудової, юридичної та іншої діяльності, керівні матеріали, книги	Книга обліку, посадова інструкція

Побудуємо концептуальну модель для системи розв'язування комбінаторно-оптимізаційних задач. Множина понять-кандидатів для цієї розробки включає в себе такі словосполучення: *завдання; тип завдання; список типів завдання; спосіб задання даних; уведення даних; вибір даних із бази; алгоритм розв'язання задачі; повнота опису; результати; дані; база даних.*

Спробуємо виокремити основні поняття і пов'язати їх між собою.

Мета основного варіанта використання системи — виконання завдання. Повний опис завдання містить: тип завдання, дані та вказівку щодо алгоритму. Із цим описом будуть пов'язані й здобуті результати. Дані можуть зберігатися в базі і вводяться туди. Опис завдання і все, що з ним пов'язано, може зберігатися в базі.

Визначимо можливі узагальнення:

- *спосіб задання даних*: уведення даних, вибір даних із бази.

- *алгоритм*: алгоритм розв'язання задачі (конкретний алгоритм розв'язання задачі). Переходимо до побудови концептуальної моделі.

Основний клас-поняття, виходячи з опису — *Завдання*. Пов'язуємо класи-поняття: *Дані, Алгоритм і Результати* [7].

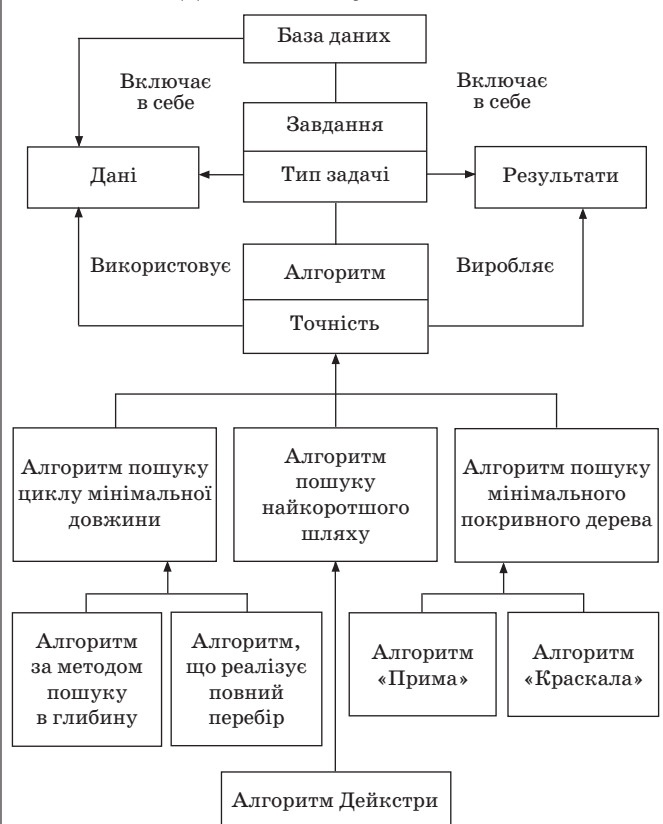
У системі управління планується реалізувати алгоритми розв'язування задач трьох типів:

- 1) пошук циклу мінімальної довжини, що проходить через усі вершини;

- 2) пошук найкоротшого шляху;

- 3) пошук мінімального покривного дерева.

Отже, клас-поняття *Алгоритм* являє собою суперклас для класів *Алгоритм пошуку циклу мінімальної довжини, Алгоритм пошуку найкоротшого шляху і Алгоритм пошуку мінімального покривного дерева* (див. рисунок). Від них, у свою чергу, будуть успадковуватися *Алгоритми*, які реалізують конкретні методи. Алгоритм також пов'язаний з *Даними і Результатами*.



Контекстна діаграма класів для системи розв'язання комбінаторно-оптимізаційних задач

Діаграми класів

Діаграма класів — центральна ланка об'єктно-орієнтованих методів розробки програмного забезпечення (ПЗ), тому всі наявні методи використовують діаграми класів в одній із відомих нотацій. Проте здебільшого діаграми класів у цих методах застосовують на етапі проектування, щоб розкрити особливості побудови конкретних класів.

Зауважимо, що мова UML, на відміну від решти нотацій, пропонує використовувати три рівні діаграми класів — залежно від ступеня їх деталізації [8]:

1) *концептуальний рівень*, на якому діаграми класів, звані в цьому разі *контекстними*, унаочнюють зв'язок між основними поняттями предметної області;

2) *рівень специфікацій*, на якому діаграми класів відбивають інтерфейси класів предметної області, тобто зв'язки об'єктів цих класів;

3) *рівень реалізації*, на якому діаграми класів безпосередньо унаочнюють поля та операції конкретних класів.

Практично йдеться про три різні моделі, зв'язок між якими неоднозначний. Так, якщо концептуальна модель визначає деякі поняття предметної області як клас, то це не означає, що для реалізації цього поняття буде використано окремий клас. Проте в усіх трьох моделях нас цікавлять типи об'єктів (класи) та їхні статичні відношення, що дозволяє використовувати єдину нотацію.

Кожну з перелічених моделей використовують на конкретному етапі розробки:

- концептуальну модель — на етапі аналізу;
- діаграми класів рівня специфікації — на етапі проектування;
- діаграми класів рівня реалізації — на етапі реалізації.

Діаграма класів описує типи об'єктів системи і різного роду статичні відношення, які існують між ними. Розглядають два основні види статичних відношень:

- *асоціації* (наприклад, клієнт може взяти на прокат ряд відеокасет);
- *ідтипи* (медсестра є різновид особистості).

На діаграмах класів зображуються також атрибути класів, операції класів і обмеження, які накладаються на зв'язки між об'єктами.

Графічно клас у нотації мови UML зображується у вигляді прямокутника, який додатково може бути поділений горизонтальними лініями на розділи або секції. У цих секціях може бути зазначено ім'я класу, атрибути та операції класу.

На початкових етапах розробки діаграми окремі класи можуть позначатися простим прямокутником, в якому має бути вказано ім'я відповідного класу. У процесі опрацювання окремих компонентів діаграми опис класів доповнюється атрибутами та операціями. Четверта секція не обов'язкова

і слугує для розміщення додаткової інформації довідкового характеру, наприклад про виключення або обмеження класу, відомості про розробника або мову реалізації. Передбачається, що остаточний варіант діаграми містить найбільш повний опис класів, які складаються з трьох або чотирьох секцій.

Динамічні, статичні моделі та моделі реалізації об'єктно-орієнтованих програмних систем

Основне завдання логічного проектування в разі об'єктного підходу є розробка класів для реалізації об'єктів, отриманих у результаті об'єктної декомпозиції, що передбачає повний опис полів і методів кожного класу.

Фізичне проектування згідно з об'єктним підходом включає в себе об'єднання класів та інших програмних ресурсів у програмні компоненти, а також розміщення цих компонентів на конкретних обчислювальних пристроях.

Статичні моделі об'єктно-орієнтованих програмних систем забезпечують подання структури систем у термінах базових будівельних блоків і відношень між ними. «Статичність» цих моделей полягає в тому, що вони не відбивають динаміки змін системи в часі. Утім слід розуміти, що ці моделі несуть у собі опис не тільки структури, а й операцій, які реалізують задану поведінку системи. Основний засіб для подання статичних моделей є діаграми класів.

Динамічні моделі забезпечують подання поведінки систем. «Динамізм» цих моделей полягає в тому, що вони відбивають зміну станів у процесі роботи системи (залежно від часу). Для моделювання поведінки системи використовують:

- *автомати* — описують поведінку в термінах послідовності станів, через які проходить об'єкт протягом свого життя;
- *взаємодії* — описують поведінку в термінах обміну повідомленнями між об'єктами.

Статичні і динамічні моделі описують логічну організацію системи, відбиваючи логіку програмного забезпечення. *Моделі реалізації* являють собою подання системи у фізичному світі, з охопленням питань упакування логічних елементів у компоненти та розміщення компонентів в апаратних вузлах. Засобом для побудови моделі реалізації виступають діаграми компонентів і діаграми розміщення.

Висновки

Концептуальна модель характеризує статичні властивості розроблюваного програмного забезпечення. Для опису особливостей його поведінки, тобто можливих дій системи, доцільно використовувати діаграми класів і діаграми станів об'єктів.

Література

1. Буч, Г. Объектно-ориентированное проектирование с примерами применения / Г. Буч; пер. с англ. — М.: Конкорд, 1992.— 519 с.
2. Волкова, В. Н. Теория систем и методы системного анализа в управлении и связи / [В. Н. Волкова, В. А. Воронков, А. А. Денисов и др.].— М.: Радио и связь, 1985.— 136 с.
3. Гинзбург, Б. М. Декомпозиционный алгоритм оптимизации распределения потоков в сети связи ЭВМ / Б. М. Гинзбург // Управление на сетях и узлах связи.— М.: Наука, 1979.— С. 65–78.
4. Жураковский, Б. Ю. Об'єктно-орієнтована модель системи управління мережею NGN / Б. Ю. Жураковський // Вісник ДУКТ.— 2012.— Т. 10, № 3.— С. 81–84.
5. Шварцман, В. О. Передача данных и концептуальная модель телеинформационных служб / В. О. Шварцман // Электросвязь.— 1995.— № 10.— С. 16–20.
6. Шлеер, С. Объектно-ориентированный анализ: моделирование мира в состояниях / С. Шлеер, С. Меллор; пер. с англ.— К.: Диалектика, 1993.— 240 с.
7. Sylaer, S. Object-oriented Systems Analysis: Modeling the world in Data / Sally Sylaer and Stephen J. Mellor.— N. J., 1988.— 145 p.
8. Sylaer, S. An Object-Oriented Approach to Domain Analysis / Sally Sylaer and Stephen J. Mellor.— N. Y., 1989.— 97 p.

Б. Ю. Жураковский

ПОСТРОЕНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ СИСТЕМЫ УПРАВЛЕНИЯ ИНФОКОММУНИКАЦИОННОЙ СЕТЬЮ

Исследованы актуальные проблемы определения объекта управления для концептуальной модели системы управления, что особенно важно при проектировании и построении системы управления. Рассмотрены основные составные части объектно-ориентированной методологии: объектно-ориентированный анализ; объектно-ориентированное проектирование; объектно-ориентированное программирование. Определены такие понятия, как диаграмма классов, статическая и динамическая модель объектно-ориентированных программных систем. Приведены варианты графического изображения класса на диаграмме классов. Предложена контекстная диаграмма классов для системы решения комбинаторно-оптимизационных задач в системе управления инфокоммуникационной сетью.

Ключевые слова: объектно-ориентированная модель; системы управления сетью; объектно-ориентированный анализ; модели состояний; диаграмма классов.

B. Yu. Zhurakovsky

CONSTRUCTION OF CONCEPTUAL MODEL SYSTEMS MANAGEMENT BY OF INFOCOMMUNICATION NETWORK

In hired the issues of the day of determination of object of management for the conceptual model of the system of management are investigational, that it is especially important at planning and construction of control system. Basic component parts of the object-oriented methodology are considered: the object-oriented analysis; object-oriented planning; object-oriented programming. A concept is certain diagram of classes, static and dynamic model of the object-oriented programmatic systems. The brought variants over of graphic image of class on the diagram of classes. The context diagram of classes is offered for the system of decision of combinatorics-optimization tasks. in control system infocommunication network.

Keywords: the object-oriented model; control system by a network; object-oriented analysis; models of the states; diagram of classes.

УДК 621.391.004.15

В. П. РОМАНЕНКО, канд. техн. наук

ФОРМАЛІЗАЦІЯ ПРОЦЕСУ РОЗРОБКИ АЛГОРИТМІВ ПАРАЛЕЛЬНОГО ПОШУКУ ДЕФЕКТІВ ЗІ ВЗАЄМНИМ ОБМІНОМ ІНФОРМАЦІЄЮ ПРО РЕЗУЛЬТАТИ ДІАГНОСТУВАННЯ

Формалізовано порядок розробки умовних алгоритмів паралельного пошуку дефектів за графічними та матричними моделями об'єктів діагностування.

Ключові слова: умовний алгоритм діагностування; граф інформаційно-енергетичних зв'язків; таблиця функцій несправностей.

Вступ

Останніми роками було виконано поглиблені дослідження умовних алгоритмів діагностування (УАД) у вигляді групових дерев логічних можливостей (ГДЛМ) для діагностування об'єктів великої розмірності з реалізацією паралельного спільного [1] і зонного [2] пошуку дефектів. Проте й досі немає науково обґрунтованих рекомендацій з організації процесу пошуку дефектів і побудови ГДЛМ оптимальної форми.

Мета статті — подати рекомендації щодо формалізації зазначених дій.