

GENERAL ISSUES OF ENGINEERING SCIENCESDOI <https://doi.org/10.30525/978-9934-26-109-1-31>**АНАЛІЗ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПІДХОДУ****Дмітрієв О. М.**

*доктор технічних наук, доцент,
завідувач кафедри льотної експлуатації, аеродинаміки
та динаміки польоту
Льотна академія
Національного авіаційного університету*

Келлер І. К.

*аспірант
Льотна академія
Національного авіаційного університету
м. Кропивницький, Україна*

Одним з найпоширеніших видів моделювання в даний час є моделювання із застосуванням різних інформаційних технологій, тобто побудова і використання комп'ютерних моделей у вигляді прикладних програм різного призначення (програмних додатків). Створення цих додатків може здійснюватися традиційними методами, заснованими на алгоритмах, процедурах і даних, а також більш сучасними – об'єктно-орієнтованими, заснованими на концептуальному моделюванні предметної області [5, с. 3].

Розглянемо основні поняття об'єктно-орієнтованої методології створення програмного забезпечення (ПЗ), що зараз нерозривно пов'язано з використанням мови UML (Unified Modeling Language – уніфікованої мови моделювання), що «являє собою систему позначень, що базується на діаграмах і призначена для моделювання систем на основі об'єктно-орієнтованого підходу» [3, с. 4].

Особливістю даної методології є програмний опис не лише процедур, необхідних для вирішення задачі, а тих сутностей, які беруть участь в цих процедурах і їх забезпечують. Справа в тому, що процедури здійснюються не самі по собі, не якимось абстрактним

чином, а шляхом взаємодії деяких цілком конкретних об'єктів, відповідно до наявних у них властивостей. І якщо ці об'єкти і їх властивості описані в програмі, то залишається тільки дати цим об'єктам можливість взаємодіяти. Процедури, які необхідно виконати для вирішення завдання, будуть здійснюватися як результат цієї взаємодії [4, с. 16].

Об'єктно-орієнтоване моделювання та розробка ПЗ здійснюються шляхом послідовного виконання:

- Об'єктно-орієнтованого аналізу (object-oriented analysis – OOA), при якому моделюється, розробляється та аналізуються системиз точки зору класів і об'єктів, виявлених в предметній області.
- Об'єктно-орієнтованого проектування (object-oriented design – OOD), при якому шляхом об'єктної декомпозиції створюється об'єктна модель.
- Об'єктно-орієнтованого програмування (object-oriented programming – OOP), при якому програма представляється у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, а класи створюють ієрархію спадкування [1, с. 78].

Відповідно до вимог OOA і OOD для побудови об'єктної моделі складної системи її необхідно представити в канонічній формі. Ця форма представлення системи включає в себе дві ортогональних ієрархії: ієрархія класів та ієрархія об'єктів [1, с. 78]. Передбачається, що така об'єктно-орієнтована декомпозиція системи, дозволяє розкрити її повну архітектуру, тобто структуру класів і структуру об'єктів.

При цьому якщо об'єкт, як екземпляр класу, відповідає конкретному предмету або явищу, визначеному в часі і в просторі, то клас – абстракції істотного в об'єкті. Таким чином, в рамках об'єктно-орієнтованого підходу клас розглядається як безліч об'єктів (екземплярів), що мають загальну структуру та поведінку. Об'єкт «являє собою конкретний упізнаваний предмет, одиницю або сутність (реальну або абстрактну), що має чітко визначене функціональне призначення в даній предметній області» [1, с. 92].

При цьому властивості (стан і поведінку) об'єктів, як екземпляр класів, в об'єктній моделі, визначає відповідний клас в ієрархії класів, яка описує предметну область, що моделюється. Головним же завданням OOA і OOD вважається «вибір правильного набору абстракцій для опису заданої предметної області» [1, с. 56], що підкреслює важливість концептуального класифікаційного моделювання в процесі об'єктно-орієнтованої розробки.

Концептуальна база методології об'єктно-орієнтованого моделювання включає в себе чотири головних взаємопов'язаних поняття [1, с. 78-84]: абстрагування, ієрархія, інкапсуляція і модульність.

- Абстрагування – спосіб виділення істотних характеристик деякого об'єкта (абстракцій), що відрізняють його від всіх інших видів об'єктів і, таким чином, чітко визначають його концептуальні межі.

- Ієрархія – спосіб упорядкування абстракцій (класів) за рівнями.

- Інкапсуляція – спосіб відділення елементів об'єкта (класу), що визначають його пристрій, від елементів, що визначають його поведінку (тобто відокремлення реалізації від інтерфейсу).

- Модульність – спосіб розкладання системи на пов'язані, але відносно самостійні частини (модулі).

Об'єктно-орієнтоване моделювання, проектування та програмування в обов'язковому порядку засноване на використанні названих понять при описі класів і об'єктів.

Опис класу включає атрибути, за допомогою яких характеризується стан об'єктів даного класу, і операції, за допомогою яких характеризується функціонування (поведінка) об'єктів даного класу. При цьому розмежовується зовнішній вигляд класу (специфікація або інтерфейс) і його внутрішній устрій (реалізація). Головне в інтерфейсі – оголошення операцій, підтримуваних екземплярами класу. У нього також можуть входити оголошення інших класів, змінних, виняткових ситуацій, що уточнюють абстракцію, яку клас повинен виражати. Реалізація, як правило, полягає у визначенні операцій, оголошених в інтерфейсі класу [4, с. 11].

Відносини між класами:

- Наслідування – таке відношення між класами, при якому клас повторює опис стану та поведінку суперкласу (одного – одиночне успадкування; кількох – множинне успадкування). Дане відношення є основним при виявленні ієрархії класів предметної області. Вузкі, спеціалізовані класи в ієрархії, від яких створюються екземпляри (об'єкти), називаються конкретними класами. Загальні класи, від яких екземпляри не виробляються, називаються абстрактними класами. Самий загальний клас в ієрархії класів називається базовим.

- Асоціація – відношення семантичної залежності, що показує які ролі класи грають один для одного. Асоціації розрізняються по потужності: «один до одного», «один до багатьох», «багато до багатьох».

- Агрегація – відповідне відношенню «частина-ціле» між об'єктами даних класів.

• Використання –розглядається як різновид відносини асоціації, при якому одна зі сторін (клієнт) користується послугами або ресурсами іншого боку (сервера). Крім того, використання може розглядатися як один з аспектів відносини спадкування, так як підклас, наслідуючи стан і поведінку класу, виступає в ролі його клієнта. Так само клієнтом класу є його примірник (об'єкт), який використовує атрибути і операції даного класу.

При цьому операція розглядається як послуга, яку клас може надати своїм клієнтам. Як правило, ці операції бувають наступних видів:

- Модифікатор – операція зміни стану об'єкта.
- Селектор – операція зчитування стану об'єкта (без зміни стану).
- Ітератор – операція з організації доступу до всіх частин об'єкта в строго визначеної послідовності.
- Конструктор – операція створення та / або ініціалізації об'єкта.
- Деструктор – операція, яка звільняє стан об'єкта та / або руйнує сам об'єкт.

Опис об'єкта включає в себе опис його стану, яке характеризується переліком атрибутів, що відповідають класу, екземпляром якого є даний об'єкт, і їх поточними значеннями. Крім того, опис об'єкта включає також опис його поведінки (функції), яке характеризується методами, що реалізують операції класу, екземпляром якого є даний об'єкт.

Зв'язки позначають рівноправні або «клієнт-серверні» відносини між об'єктами [5, с. 28].

Беручи участь в зв'язках, об'єкт може виконувати одну з трьох ролей:

- Актор – при якій об'єкт може впливати на інші об'єкти, але сам ніколи не піддається їх впливу (активний об'єкт);
- Сервер – за якої об'єкт може тільки піддаватися впливу з боку інших об'єктів, але ніколи не виступає в ролі впливає об'єкта (пасивний об'єкт);
- Агент – при якій об'єкт може бути і активним, і пасивним.

Крім того, між об'єктами можуть бути виявлені ієрархічні відносини, а саме ставлення «частина-ціле», тобто агрегація. При цьому, йдучи від цілого (агрегату), можна прийти до його частин (атрибутів). Агрегація означає фізичне або концептуальне входження одного об'єкта в інший.

Нааявність різних варіантів і відсутність будь-яких обґрунтувань цих варіантів свідчить про те, що на сьогоднішній день системні методи

класифікації, що дозволяють обґрунтовано виділяти класи і об'єкти, в об'єктно-орієнтованій технології поки ще не застосовуються у достатній кількості. Даний підхід може бути використаний для опису процесу прийняття рішень оператором складних систем управління (пілот літака тощо).

Література:

1. Буч Г. Объектно-ориентированный анализ и проектирование. Москва, «Бином», 1998. 356 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд. Москва: «Бином», 1998. 560 с.
3. Larman С. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition. London, «Pearson», 2004. 736 p.
4. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition –Boston, «Addison-Wesley Professional», 2003. 208 p.
5. McLaughlin B.D., Pollice G., West D. Head First Object-Oriented Analysis and Design, 1st Edition – Sebastopol, «O'Reilly Media», 2006. 636 p.