

Секція

«ЕКОНОМІКА ПІДПРИЄМСТВА»

ГОЛОВА СЕКЦІЇ – д.е.н., професор І.А. Дмитрієв

УДК 625.317/2

ТЕОРЕТИЧНІ ПІДХОДИ ЩОДО ПРОБЛЕМИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ПІДПРИЄМСТВА

*Басова Д.В., ст.. викладач,
Лакуста Г.Г аспірант (УкрДУЗТ)*

Структурна реформа підприємств залізничного транспорту потребує в найкоротші строки вирішення завдань по оновленню технічних засобів та підвищенню ефективності роботи галузі на основі інформатизації, впровадження нових технологій, використання нових засобів зв'язку. В таких умовах особливого значення набувають питання розробки та впровадження на підприємствах транспорту ресурсозберігаючих технологій, що дозволить вивільнити частину засобів на першочергову модернізацію транспортної техніки. Тому актуальним є розробка пропозицій які забезпечать вирішення цієї проблеми

Описання нової транспортної системи залізничного транспорту - це робота з прямого інжинірингу, що починається з формулювання цілей і образу майбутньої, створюється загальне описання процесу, що включає замовників, постачальників, а також самий процес. Далі проводиться імітаційне моделювання різних процесів. Нарешті, обрана альтернатива реалізується.

Методики, засновані на моделюванні інформаційних систем, виходять із наступної парадигми. При описанні інформаційної системи передбачається, що вона містить, два типи сутностей: деякий аналог програми (операційні сутності, що виконують деяку обробку) і дані (пасивні сутності, що беруть інформацію, доступну для пошуку, читання і заміни). Іншими словами, інформаційна система описується як деяка абстракція комп'ютера. При моделюванні (розробці) складні інформаційні системи розбиваються на складові частини, кожна з яких розглядається окремо від інших. Такий

прийом називається декомпозицією. Класичний підхід до розробки складних систем являє собою структурне проектування, при якому здійснюється алгоритмічна декомпозиція системи по методу «зверху - вниз».

Життєвий цикл розробки складної системи складається з етапів аналізу, програмування, тестування і супроводу, що виконуються послідовно. Такий метод, названий каскадним, має наступні особливості: лінійність виконання етапів життєвого циклу розробки; чіткий поділ даних і процесів їхньої обробки; використання процедурних мов програмування. Недолік - послідовне виконання етапів, що приводить до великих втрат часу, не дозволяє швидко розробляти прототипи програмної системи. Каскадний принцип не погодиться з ітеративним характером розробки програмної системи, оскільки на останніх етапах може з'ясуватися необхідність внесення змін у рішення, прийняті на попередніх етапах.

Для усунення цього недоліку запропоновано спіральний підхід. Він полягає в тому, що розробка проекту ведеться якби по спіралі, причому на кожному її витку виконуються послідовно перераховані вище етапи, на яких уточнюється проект. Цей підхід доповнює каскадний метод елементами інтерактивності. Але і для нього характерний ряд істотних недоліків: трудомісткість внесення змін, великий об'єм документації по проекту, що утруднює програмування; серйозні обмеження можливості і складання системи з готових компонентів; складність переносу на інші платформи.

У розроблювача предметної області в якості об'єктів можуть розглядатися конкретні цілі, а також абстрактні або реальні сутності. Кожен об'єкт є представником деякого класу однотипних об'єктів. Клас визначає загальні властивості для всіх його об'єктів. До таких властивостей відносяться:

- склад і структура даних, що описують атрибути класу і відповідних об'єктів;

- сукупність методів - процедур, що визначають взаємодію об'єктів цього класу з

зовнішнім середовищем.

Інкапсуляція - приховування інформації. При об'єктно-орієнтованому програмуванні передбачена можливість заборонити будь-який доступ до атрибутів об'єктів, крім як через його методи. Внутрішня структура об'єкта в цьому випадку схована від користувача, тобто об'єкти можна вважати самостійними сутностями, відділеними від зовнішнього світу. Для того щоб об'єкт зробив деяку дію, йому ззовні необхідно послати повідомлення, що ініціює виконання потрібного методу. Інкапсуляція дозволяє змінювати реалізацію будь-якого класу об'єктів без побоювання, що це викличе небажані побічні ефекти в програмній системі. Тим самим спрощується процес виправлення помилок і модифікації програм.

Спадкування - можливість створювати з класів нові класи за принципом «від загального до приватного». Спадкування дозволяє новим класам при збереженні властивостей класів-батьків додавати свої риси, що відбивають їхню індивідуальність. Спадкування дозволяє створювати ієрархії класів і є ефективним засобом внесення змін і доповнень у програмні системи.

Поліморфізм - здатність об'єктів вибирати метод на основі типів даних, застосовуваних у повідомленні. Кожен об'єкт може реагувати по-своєму на те саме повідомлення. Поліморфізм дозволяє спростити вихідні тексти програм, забезпечує їх розвиток за рахунок введення нових методів обробки.

Об'єктно-орієнтована декомпозиція полягає в представленні системи у виді сукупності класів і об'єктів предметної області. При цьому ієрархічний характер складної системи відбивається у виді ієрархії класів, а її функціонування розглядається як взаємодія об'єктів.

Життєвий цикл об'єктно-орієнтованої розробки програмних систем містить декілька етапів, але на відміну від структурного підходу в ньому немає суворої послідовності їхнього виконання. Процес носить принципово ітеративний характер, що цілком відповідає потребам розроблювачів. Після обстеження починається об'єктно-орієнтоване проектування, у ході якого деталізується представлення класів і об'єктів, отриманих на етапі аналізу. Визначаються структури даних,

методи, відношення між класами, розробляються сценарії взаємодії об'єктів. При проектуванні системи можуть вводитися нові класи й об'єкти, якщо це буде потрібно для рішення поставлених проблем. У результаті проектування повинна бути створена детальна модель системи, складена специфікація об'єктів, класів і відношень, достатня для її програмування.

Об'єктно-орієнтований підхід забезпечує швидке створення прототипів проектованої системи, поступовий розвиток яких приводить до кінцевого результату. На цьому етапі також можливо введення нових класів, зміна структур даних, додавання нових методів. При даному підході модифікація не вимагає повного перегляду проекту, торкаючись лише необхідних для цього класів й об'єктів.

Головна особливість життєвого циклу при об'єктно-орієнтованому підході полягає в тому, що немає суворої послідовності виконання окремих етапів. При розробці може з'ясуватися необхідність додаткового обстеження, програмування і наступне тестування можуть зажадати повернення до проектування. Такий метод відбиває ітеративний характер розробки додатка.

Об'єктно-орієнтовані технології не дають негайної віддачі. Ефект від їхнього застосування починає позначатися після розробки двох-трьох проектів і нагромадження повторно використовуваних компонентів, що відбивають типові проектні рішення в даній області.

Висновок. Директива звичайно ініціює зусилля по реінжинірингу яка складена в термінах високого рівня, її загальний тон повинен виражати чекання від реалізації проекту. Щоб стимулювати істотні зміни в транспортному промисловому виробництві, приділяється увага серйозним проблемам і показуються, які радикальні зміни очікуються в майбутньому.