

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ UML

Робота присвячується аналізу проектування програмного забезпечення (ПЗ) на базі UML (англ. Unified Modeling Language - уніфікована мова моделювання) - мова графічного опису для об'єктного моделювання в області розробки ПЗ, моделювання бізнес-процесів, системного проектування та відображення організаційних структур. **UML забезпечує підтримку всіх семи ступенів проектування ПЗ:** а) моделювання бізнес-прецедентів, б) розробка моделі бізнес-об'єктів, в) розробка концептуальної моделі даних, г) розробка вимог до системи, д) аналіз вимог і попереднє проектування системи, е) розробка моделей бази даних і додатків, ж) проектування фізичної реалізації системи життєвого циклу ПЗ і надає для цих цілей ряд графічних засобів.

На ступені проектування при створенні **концептуальної моделі** для опису **бізнес-діяльності** використовуються моделі **бізнес-прецедентів** і **діаграми видів діяльності**, для опису бізнес-об'єктів - моделі бізнес-об'єктів і **діаграми послідовностей**.

Бізнес-діяльність включає два види: **Модель бізнес-прецедентів** описує бізнес-процеси з точки зору зовнішнього користувача, тобто відображає погляд на діяльність організації ззовні. **Діаграми видів діяльності** - UML-діаграма, на якій показується розкладання деякої діяльності на її складові частини. Під діяльністю розуміється специфікація виконуваного поведінки у вигляді координованого послідовного і паралельного виконання підлеглих елементів-вкладених видів діяльності і окремих дій з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів іншого. **Діаграми діяльності** використовуються при моделюванні бізнес-процесів, технологічних процесів, послідовних і паралельних обчислень. **Діаграми діяльності** складаються з обмеженої кількості фігур, з'єднаних стрілками. Основні фігури: прямокутники з заокругленнями - **дії**; ромби - **рішення**; широкі смуги - **початок (розгалуження) і закінчення (сходження) розгалуження дії**; чорний коло - **початок процесу (початковий стан)**; Чорний круг з обвідкою - **закінчення процесу (кінцевий стан)**. Стрілки йдуть від початку до кінця процесу і показують послідовність переходів.

Бізнес-об'єкти представляються: **Моделлю бізнес - об'єктів**, яка показує виконання бізнес-процесів організації її внутрішніми виконавцями. Основними компонентами моделей бізнес-об'єктів є зовнішні і внутрішні виконавці, а також бізнес-суті, відображають все, що використовують внутрішні виконавці для реалізації бізнес-процесів. **Діаграмами послідовностей**, на якій для деякого набору об'єктів на єдиній тимчасовій осі показані життєвий цикл (*створення - діяльність - знищення*) і взаємодія (відправка запитів і отримання відповідей). Основними елементами діаграми послідовності є позначення об'єктів (прямокутники з назвами об'єктів), вертикальні «лінії життя» (англ. Lifeline), що відображають плин часу, прямокутники, що відображають діяльність об'єкта або виконання ним певної функції (прямокутники на пунктирною «лінії життя»), і стрілки, що показують обмін сигналами або повідомленнями між об'єктами.

На ступені, при створенні логічної моделі ІС, опис вимог до системи задається у вигляді моделі опису **системних прецедентів**, а попереднє проектування здійснюється з використанням діаграм класів, діаграм послідовностей і діаграм станів. **Логічна модель** описує поняття предметної області, їх взаємозв'язок, а також обмеження на дані, що накладаються предметною областю. Приклади понять - "співробітник", "відділ", "проект", "зарплата". Приклади взаємозв'язків між поняттями - "співробітник числиться рівно в одному відділі", "співробітник може виконувати кілька проектів", "над одним проектом може працювати декілька співробітників". Приклади обмежень - "вік співробітника не менше 16 і не більше 60 років". Модель опису системних прецедентів, відображає виконання конкретних обов'язків внутрішніми і зовнішніми виконавцями з використанням ІС. Джерелом даних для створення моделі системних прецедентів є розроблені на попередньому етапі бізнес-моделі. Однак при створенні моделі корисно попередньо скласти детальні описи прецедентів, які містять визначення використовуваних даних і точну послідовність їх виконання. **Прецедент** - закінчена послідовність дій, ініційована зовнішнім об'єктом (особистістю або системою), яка взаємодіє з ІС і отримує в результаті деяке повідомлення від ІС. На діаграмі представляється овалом з написом, що відображає зміст дії

На ступені створення фізичної моделі, детальне проектування виконується з використанням: **діаграм класів, діаграм компонентів і діаграм розгортання**. Розглянемо ці діаграми, та проведемо аналіз, в порядку написання

Діаграма класів - це діаграма, що демонструє класи системи, їх атрибути, методи і взаємозв'язку між ними. Вони є однією з форм статичного опису системи з точки зору її проектування, показуючи її структуру. Діаграма класів не відображає динамічну поведінку об'єктів зображених на ній класів. На діаграмах класів показуються класи, інтерфейси і відносини між ними. Клас - це основний будівельний блок ПС. Це поняття присутній і в ОО мовах програмування, тобто між класами UML і програмними класами є відповідність, що є основою для автоматичної генерації програмних кодів або для виконання реінжинірингу. Кожен клас має назву, атрибути і операції. Клас на діаграмі показується у вигляді прямокутника, розділеного на 3 області. У верхній міститься назва класу, в середній - опис атрибутів (властивостей), в нижній - назви операцій - послуг, що надаються об'єктами цього класу. Діаграма класів є подальшим розвитком концептуальної моделі спроектованої системи.

Діаграма компонентів, статична структурна діаграма, показує розбиття програмної системи на структурні компоненти та зв'язку (залежності) між компонентами. Як фізичних компонентів можуть виступати файли, бібліотеки, модулі, виконувані файли, пакети і так далі. Діаграма компонентів дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний, бінарний і виконуваний код. У багатьох середовищах розробки модуль або компонент відповідає файлу. Пунктирні стрілки, що з'єднують модулі, показують відношення взаємозалежності, аналогічні тим, які мають місце при компіляції початкового програмного коду. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними. **Діаграма компонентів розробляється для наступних 4-ох цілей:** а) візуалізації загальної структури вихідного коду програмної системи; б) специфікації виконуючого варіанта програмної системи; в) забезпечення багаторазового використання окремих фрагментів програмного коду; г) подання концептуальної і фізичної схеми баз даних. У розробці діаграм компонентів беруть участь як системні аналітики та архітектори, так і програмісти. Діаграма компонентів забезпечує узгоджений перехід від логічного представлення до конкретної реалізації проекту у формі програмного коду. Одні компоненти можуть існувати тільки на етапі компіляції програмного коду, інші - на етапі його виконання. Діаграма компонентів відображає загальні залежності між компонентами, розглядаючи останні як класифікаторів.

Діаграма розгортання, або застосування, - це один з двох видів діаграм, що використовуються при моделюванні фізичних аспектів об'єктно-орієнтованої системи. Діаграма розгортання призначена для візуалізації елементів і компонентів програми, існуючих лише на етапі її виконання. При цьому подаються тільки компоненти - екземпляри програми, які є виконуваними файлами або динамічними бібліотеками. Ті компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не показуються. Діаграма розгортання розробляється спільно системними аналітиками, мережевими

інженерами і системотехніками.

На рис. 1 показані відносини між різними видами діаграм UML. Показники стрілок можна інтерпретувати як відношення «є джерелом вхідних даних для ...» (наприклад, діаграма класів є джерелом даних для п'яти діаграм: компонентів, станів, компаративні, видів послідовності і прецедентів).

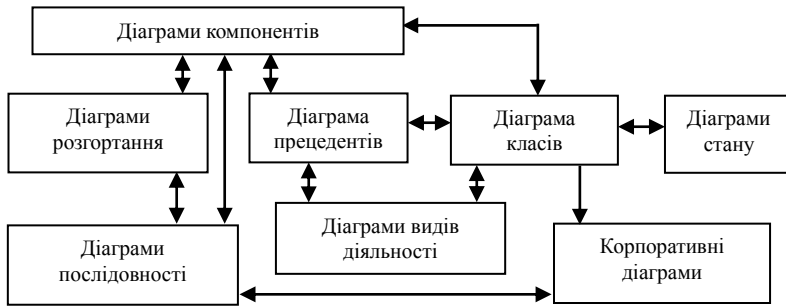


Рис. 1 Взаємозв'язок між діаграмами UML

Наведена схема є наочною ілюстрацією ітеративного характеру розробки моделей з використанням UML.

Висновок У даній роботі відображені концептуальні погляди авторів на аналіз проектування ПЗ на базі UML, з врахуванням обмежень.