

Кафедра управління інформаційною безпекою

Спеціальність «Кібербезпека», 5 курс навчання

ЛЕКЦІЯ. ЗАСОБИ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ БАЗ ДАНИХ

План:

1. CASE-технології.
2. RAD-технології та компоненто-орієнтовані технології.

Література

1. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. К.: КНУБА, 2005. – 204 с.
2. Гайна Г.А. Організація баз даних і знань. Мови баз даних: Конспект лекцій.–К.:КНУБА, 2002. – 64 с.
3. Гайна Г.А., Попович Н.Л. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій.–К.:КНУБА, 2000. – 76 с.

1. CASE-технології.

Проблеми автоматизації проектування інформаційних систем викликали необхідність в програмно-технологічних засобах спеціального класу – CASE-засобах (Computer Aided Software Engineering). CASE-технологія являє собою сукупність методів проектування програмного забезпечення, а також набір інструментальних засобів, які дозволяють у наочній формі моделювати предметну область, аналізувати цю модель на всіх стадіях розробки і супроводження програмного забезпечення і розробляти застосування згідно з інформаційними потребами користувачів.

Методологія CASE-технологій ґрунтується на спадному підході до проектування і дає змогу стежити за всіма етапами життєвого циклу інформаційної системи або її окремих задач. Сутність спадного підходу до проектування полягає в тому, що під час реалізації системи її характеристики конкретизуються все більше і більше.

Переваги від застосування CASE-технологій при проектуванні інформаційних систем полягають у такому:

– прискорюється та полегшується процес розробки, підвищується якість розроблюваних інформаційних систем;

– з'являється можливість переносу застосувань із середовища однієї СУБД в іншу за рахунок перетворення концептуальної моделі на фізичну і навпаки;

– з'являється можливість проведення більш досконалого моделювання системи на початкових етапах розробки.

У табл. 1 наведено, порівняння якісних змін процесу розробки інформаційної системи при переході до використання CASE-засобів.

Таблиця 1

Порівняння трудомісткості традиційної розробки
інформаційної системи і розробки із застосуванням
CASE-засобів

Традиційна розробка	Розробка з використанням CASE-засобів
Основні зусилля на кодування і тестування	Основні зусилля на аналіз і проектування
«Паперові» специфікації	Швидке ітераційне прототипування
Ручне кодування	Автоматична генерація коду
Ручне документування	Автоматична генерація документації
Тестування кодів	Автоматичний контроль проекту
Супроводження кодів	Супроводження специфікації проектування

Одним з напрямів застосування CASE-засобів є інформаційне проектування, яке реалізує методики проектування БД. Крім того CASE-засоби забезпечують проектування застосувань.

Інструментальні CASE-засоби складаються з таких компонентів:

- графічні засоби, які призначені для розробки структурних діаграм;
- блоки дизайну і створення звітів для розробки інформаційних форматів вводу/виводу й інтерфейсів користувача;
- інтегрований архів (репозиторій) для збереження даних проекту системи зі словником даних;
- генератор програмної документації;
- засоби для забезпечення повністю автоматизованої перевірки несуперечливості системи, її синтаксису і повноти.

В базі даних CASE-системи зберігаються дані, які відносяться до різних етапів життєвого циклу розробки програмного забезпечення: планування, збору і аналізу вимог, проектування, реалізації, тестування, супроводження і документування.

В середовищі розробки CASE проектувальники БД і застосунки використовують CASE-інструментарій для зберігання опису схеми БД, елементів даних, прикладних процесів, екранів, звітів та інших даних, які зв'язані з процесом розробки. CASE-інструментарій інтегрує всю інформацію розробки системи в єдиному репозиторії, який адміністратор БД може перевіряти на несуперечливість і точність, і у разі потреби, виконувати виправлення.

Репозиторій – застосування БД, яке забезпечує зберігання і обробку даних і метаданих, а також їх представлення по запитах. Репозиторії можуть підтримувати різні представлення даних і метаданих, що зберігаються, множини їх версій, відображати еволюцію структури даних і метаданих.

Метадані – дані про дані, які описують їх склад і структуру, формат представлення, методи доступу і необхідні для цього повноваження користувачів, місце зберігання, семантику даних і т.ін.

Метадані виконують такі функції:

- описують властивості інформаційної системи, її механізми й інформаційні ресурси в CASE-середовищах;
- використовуються для обміну відомостями між різними інструментами CASE і/або застосуваннями інформаційної системи;
- є джерелами відомостей про властивості і зміст інформаційних ресурсів для механізмів управління даними в інформаційних системах;
- забезпечують механізми інтеграції інформаційних ресурсів з різних джерел відомостями про властивості цих ресурсів;
- є джерелом інформації, яка необхідна для перебудови інформаційних систем;
- забезпечують представлення відомостей про систему, її ресурси для різних застосунків і користувачів.

Крім того CASE-інструментарій розширює і покращує якість взаємодії між адміністратором БД, прикладними програмістами і користувачами. Адміністратор БД за допомогою CASE-інструментарію може перевіряти схеми даних для застосунків, стежити за виконанням умов про найменування,

дублювання елементів даних, перевіряти використання правил для елементів даних. CASE-засоби дозволяють каскадно передавати виправлення по всій інфраструктурі застосування, що значно спрощує роботу з впровадження системи БД.

Серед інструментальних CASE-систем розрізняють інтегровані комплекси інструментальних засобів для автоматизації всіх етапів життєвого циклу інформаційної системи (Workbench) і спеціалізовані інструментальні засоби для виконання окремих функцій (Tools).

Сучасні CASE-системи є або структурними, або об'єктно-орієнтованими.

У структурному підході до аналізу та проектування застосовуються такі види моделей:

- DFD (Data Flow Diagrams) – діаграми потоків даних;
- SADT (Structured Analysis and Design Technique – метод структурного аналізу і проектування) – моделі і відповідні функціональні діаграми;
- ERD (Entity-Relationships Diagrams) – діаграми "суть–зв'язок".

Діаграми потоків даних є основним засобом моделювання функціональних вимог до системи, що проектується. Ці вимоги представляються у вигляді ієрархії функціональних компонентів, які зв'язані потоками даних (рис. 1).

Специфікація процесів представляється у вигляді текстового опису, схем алгоритмів, псевдокодів і т.ін. *Словник* термінів являє собою короткий опис основних понять, які використовуються при створенні специфікації. *Діаграма переходів станів* демонструє поведінку системи, що розробляється. *Моделювання даних* виконується за допомогою ER-діаграм. Головна мета такого представлення – продемонструвати, як кожен процес перетворює свої вхідні дані у вихідні, а також виявити зв'язки між цими процесами.

Функціональні моделі SADT призначені для опису функціональної структури системи, що проектується.

Крім DFD для функціонального структурного і потокового моделювання застосовуються методики IDEF0 і IDEF3

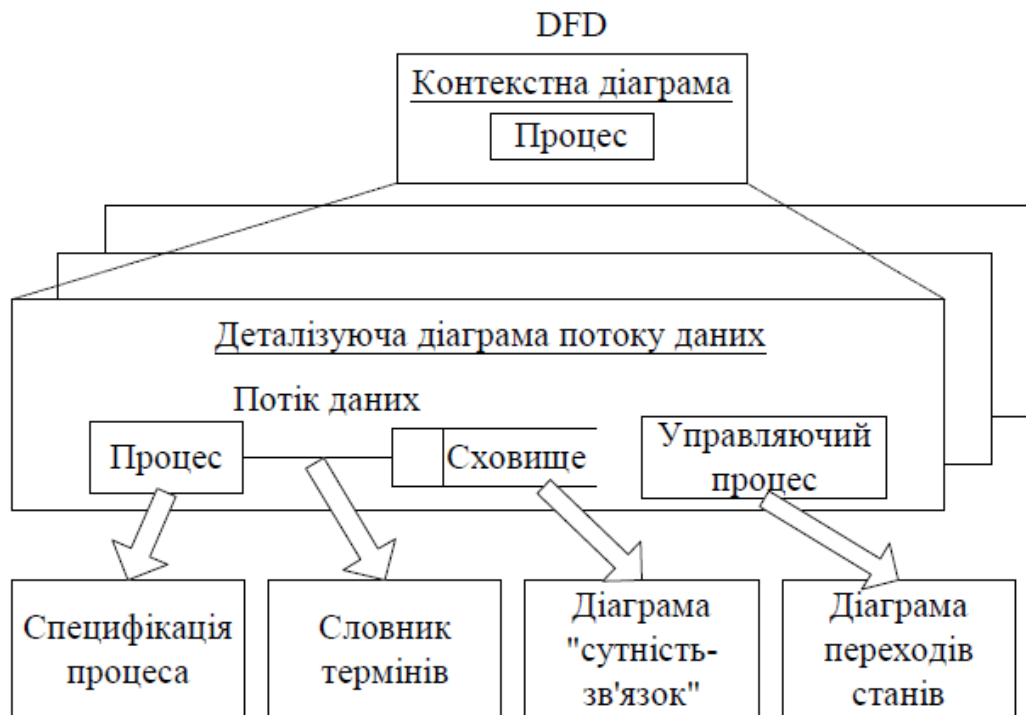


Рис. 1. Елементи специфікації методологій структурного аналізу і проектування інформаційних систем, які засновані на потоках даних

Серед *об'єктно-орієнтованих моделей* найбільш відомими є моделі побудовані за допомогою мови моделювання UML (Unified Modeling Language – уніфікована мова моделювання). Словник UML утворюють предмети, відношення, діаграми. *Предмети* розрізняють структурні, поведінки, групуючі і пояснюючі. *Відношення* існують таких видів: залежності, асоціації, узагальнення і реалізації. Проект інформаційної системи, який створюється за допомогою UML складається з діаграм: прецедентів використання, класів, станів, активності, слідування, співробітництва, компонентів, розміщення.

Основними компонентами об'єктно-орієнтованої CASE- системи є такі:

- репозиторій, який представляє об'єктно-орієнтовану БД;
- графічний інтерфейс користувача;
- засоби перегляду проекту, які дозволяють переміщуватися по елементах проекту, в тому числі по ієрархії класів і підсистем, переключення між видами діаграм;
- засоби контролю проекту;
- засоби збору статистики;

– генератор документів, який дозволяє формувати тексти вихідних документів на основі інформації з репозиторія.

Об'єктний підхід одночасно є і структурним, оскільки задовольняє основним його критеріям (розбиття на чорні скриньки, ієрархія, графічна нотация). Він також використовує діаграми потоків даних, діаграми "сутність-зв'язок", діаграми переходів станів.

Засоби моделювання даних інтегруються з іншими моделями проектування (діаграми потоків даних, об'єктно-орієнтоване моделювання і т.ін.) на основі розширених можливостей репозиторіїв, а також принципів і середовищ інтеграції CASE.

2. RAD-технології та компонентно-орієнтовані технології.

Для прискорення розробки застосувань використовуються RAD-технології. Головними рисами RAD (Rapid Application Development, середовища швидкої розробки застосувань) є такі:

- наявність об'єктно-орієнтованої мови програмування;
- візуальні засоби розробки;
- підтримка стандартних протоколів обміну даними між застосуваннями, що дозволяє розробляти багаторівневі застосування, які не залежать від джерела даних.

Ця технологія орієнтована на максимально швидке отримання перших версій програмного продукту що розробляється. У разі такого підходу систему поділяють на підсистеми, які є слабо зв'язаними за даними і функціями і точно визначають інтерфейси між різними частинами.

Процес розробки поділяється на такі етапи: аналіз і планування вимог користувачів, проектування, реалізація і впровадження.

На етапі аналізу і проектування формулюються найбільш пріоритетні вимоги, що обмежує розмір проекту. На етапі проектування застосовуються CASE-засоби. На етапі реалізації виконується ітеративна побудова реальної системи. Для контролю над виконанням вимог до системи залучаються користувачі.

Компонентно-орієнтовані технології засновані на використанні попередньо розроблених готових програмних компонентів. Тут широко застосовуються бібліотеки класів. Включення готового модуля в систему виконується за допомогою його інтерфейсу. Специфікації, які визначають інтерфейс, відокремлені від модуля, а внутрішні деталі приховані від користувача. Компоненти постачаються у скомпільованому вигляді. Звернення до модуля можливо тільки через його інтерфейс.

Користувач звертається із запитом на виконання деякої процедури. Запит відправляється посереднику. У посередника є попередньо сформований каталог (реєстр або репозиторій) інтерфейсів процедур з покажчиком на компоненти-виконавці. Після виконання процедури отримані результати повертаються користувачу.

До найбільш відомих компонентно-орієнтованих технологій належать: CORBA, COM (DCOM), JavaBeans. В якості мови інтерфейса в технологіях CORBA і DCOM використовується мова IDL (Interface Definition Language). Всі об'єкти згруповані у класи, кожний клас має свій ідентифікатор, кожний інтерфейс класа має також свій ідентифікатор. Класи об'єктів реалізуються у певному середовищі. Компоненти системного середовища об'єднані у декілька сценаріїв (потоків процедур або маршрутів), у яких виділяються точки входу для вставлення специфічних користувацьких фрагментів і розширень. Існує можливість не тільки вставляти нові фрагменти, але і замінювати вихідні компоненти в потоках процедур зі збереженням інтерфейсу.

Контрольні запитання

1. Дати визначення CASE-засобам і CASE-технологіям.
2. У чому полягають головні переваги CASE-технологій в розробці інформаційних систем?
3. Що являє собою методологія функціонального моделювання?
4. Дати визначення RAD-технології.
5. Дати визначення компонентно-орієнтованої технології.

6. Порівняти CASE-системи, які базуються на структурному і об'єктно-орієнтованому підходах.
7. Що таке репозиторій і яка його роль у інтеграції інформації?
8. Назвати сучасні CASE-системи і дати їм характеристику.

Розробив: к.т.н., доц. Полотай О.І.