***Лекція***

***Методології моделювання предметної області***

Структурна модель предметної області

В основі проектування ІС лежить моделювання предметної області. Для того щоб отримати адекватний предметної області проект ІС у вигляді системи правильно працюючих програм, необхідно мати цілісне, системне уявлення моделі, яке відображає всі аспекти функціонування майбутньої інформаційної системи. При цьому під моделлю предметної області розуміється деяка система, що імітує структуру або функціонування досліджуваної предметної області і відповідає основній вимозі - бути адекватною цій галузі.

Попереднє моделювання предметної області дозволяє скоротити час і терміни проведення проектувальних робіт і отримати більш ефективний і якісний проект. Без проведення моделювання предметної області велика ймовірність допущення великої кількості помилок у вирішенні стратегічних питань, що призводять до економічних втрат і високих витрат на подальше перепроектування системи. Внаслідок цього всі сучасні технології проектування ІС ґрунтуються на використанні методології моделювання предметної області.

До моделей предметних областей ставляться такі вимоги:

формалізація, що забезпечує однозначне опис структури предметної області;

зрозумілість для замовників і розробників на основі застосування графічних засобів відображення моделі;

реалізація, що має на увазі наявність засобів фізичної реалізації моделі предметної області в ІС;

забезпечення оцінки ефективності реалізації моделі предметної області на основі певних методів і обчислюваних показників.

Для реалізації перерахованих вимог, як правило, будується система моделей, яка відображає структурний і оцінний аспекти функціонування предметної області.

Структурний аспект передбачає побудову:

об'єктної структури, що відбиває склад взаємодіючих в процесах матеріальних та інформаційних об'єктів предметної області;

функціональної структури, що відбиває взаємозв'язок функцій (дій) щодо перетворення об'єктів в процесах;

структури управління, що відображає події та бізнес-правила, які впливають на виконання процесів;

організаційної структури, що відбиває взаємодію організаційних одиниць підприємства і персоналу в процесах;

технічної структури, яка описує топологію розташування і способи комунікації комплексу технічних засобів.

Для відображення структурного аспекту моделей предметних областей в основному використовуються графічні методи, які повинні гарантувати представлення інформації про компонентах системи. Головна вимога до графічних методів документування - простота. Графічні методи повинні забезпечувати можливість структурної декомпозиції специфікацій системи з максимальним ступенем деталізації і погоджень описів на суміжних рівнях декомпозиції.

З моделюванням безпосередньо пов'язана проблема вибору мови уявлення проектних рішень, що дозволяє якомога більше залучати майбутніх користувачів системи до її розробки. Мова моделювання - це нотація, в основному графічна, яка використовується для опису проектів. Нотація є сукупність графічних об'єктів, які використовуються в моделі. Нотація є синтаксисом мови моделювання. Мова моделювання, з одного боку, повинен виконувати рішення проектувальників зрозумілими користувачеві, з іншого боку, надавати проектувальникам кошти досить формалізованого і однозначного визначення проектних рішень, що підлягають реалізації у вигляді програмних комплексів, що утворюють цілісну систему програмного забезпечення.

Графічне зображення нерідко виявляється найбільш ємною формою подання інформації. При цьому проектувальники повинні враховувати, що графічні методи документування не можуть повністю забезпечити декомпозицію проектних рішень від постановки задачі проектування до реалізації програм ЕОМ. Труднощі виникають при переході від етапу аналізу системи до етапу проектування і особливо до програмування.

Головний критерій адекватності структурної моделі предметної області полягає у функціональній повноті розробляється ІС.

Оціночні аспекти моделювання предметної області пов'язані з розробляються показниками ефективності автоматизованих процесів, до яких відносяться:

час вирішення завдань;

вартісні витрати на обробку даних;

надійність процесів;

непрямі показники ефективності, такі, як обсяги виробництва, продуктивність праці, оборотність капіталу, рентабельність і т.д.

Для розрахунку показників ефективності, як правило, використовуються статичні методи функціонально-вартісного аналізу (ABC) і динамічні методи імітаційного моделювання.

В основі різних методологій моделювання предметної області ІВ лежать принципи послідовної деталізації абстрактних категорій. Зазвичай моделі будуються на трьох рівнях: на зовнішньому рівні (визначенні вимог), на концептуальному рівні (специфікації вимог) і внутрішньому рівні (реалізації вимог). Так, на зовнішньому рівні модель відповідає на питання, що повинна робити система, тобто визначається склад основних компонентів системи: об'єктів, функцій, подій, організаційних одиниць, технічних засобів. На концептуальному рівні модель відповідає на питання, як повинна функціонувати система? Інакше кажучи, визначається характер взаємодії компонентів системи одного і різних типів. На внутрішньому рівні модель відповідає на питання: за допомогою яких програмно-технічних засобів реалізуються вимоги до системи? З позиції життєвого циклу ІС описані рівні моделей відповідно будуються на етапах аналізу вимог, логічного (технічного) та фізичного (робочого) проектування. Розглянемо особливості побудови моделей предметної області на трьох рівнях деталізації.

об'єктна структура

Об'єкт - це сутність, яка використовується при виконанні деякої функції або операції (перетворення, обробки, формування і т.д.). Об'єкти можуть мати динамічну або статичну природу: динамічні об'єкти використовуються в одному циклі відтворення, наприклад замовлення на продукцію, рахунки на оплату, платежі; статичні об'єкти використовуються у багатьох циклах відтворення, наприклад, обладнання, персонал, запаси матеріалів.

На зовнішньому рівні деталізації моделі виділяються основні види матеріальних об'єктів (наприклад, сировину і матеріали, напівфабрикати, готові вироби, послуги) та основні види інформаційних об'єктів або документів (наприклад, замовлення, накладні, рахунки і т.д.).

На концептуальному рівні побудови моделі предметної області уточнюється склад класів об'єктів, визначаються їх атрибути і взаємозв'язку. Таким чином будується узагальнене уявлення структури предметної області.

Далі концептуальна модель на внутрішньому рівні відображається у вигляді файлів бази даних, вхідних і вихідних документів ЕІС. Причому динамічні об'єкти представляються одиницями змінної інформації або документами, а статичні об'єкти - одиницями умовно-постійної інформації у вигляді списків, номенклатур, цінників, довідників, класифікаторів. Модель бази даних як постійно підтримуваного інформаційного ресурсу відображає зберігання умовно-постійної і накопичуваної змінної інформації, використовуваної в повторюваних інформаційних процесах.

функціональна структура

Функція (операція) являє собою деякий перетворювач вхідних об'єктів у вихідні. Послідовність взаємопов'язаних по входах і виходах функцій становить бізнес-процес. Функція бізнес-процесу може породжувати об'єкти будь-якої природи (матеріальні, фінансові, інформаційні). Причому бізнес-процеси та інформаційні процеси, як правило, нерозривні, тобто функції матеріального процесу не можуть здійснюватися без інформаційної підтримки. Наприклад, відвантаження готової продукції здійснюється на основі документа "Замовлення", який, в свою чергу, породжує документ "Накладна", який супроводжує партію відвантаженого товару.

Функція може бути представлена ​​однією дією або деякою сукупністю дій. В останньому випадку кожної функції може відповідати певний процес, в якому можуть існувати свої підпроцеси, і т.д., поки кожна з підфункцій не становитиме деяку недекомпозіруемую послідовність дій.

На зовнішньому рівні моделювання визначається список основних бізнес-функцій або видів бізнес-процесів. Зазвичай таких функцій налічується 15-20.

На концептуальному рівні виділені функції декомпозіруются і будуються ієрархії взаємозалежних функцій.

На внутрішньому рівні відображається структура інформаційного процесу в комп'ютері: визначаються ієрархічні структури програмних модулів, що реалізують автоматизує функції.

Структура управління

У сукупності функцій бізнес-процесу можливі альтернативні або циклічні послідовності в залежності від різних умов протікання процесу. Ці умови пов'язані з подіями, що відбуваються у зовнішньому середовищі або в самих процесах і з освітою певних станів об'єктів (наприклад, замовлення прийняте, відкинутий, відправлений на коригування). Події викликають виконання функцій, які, в свою чергу, змінюють стану об'єктів і формують нові події, і т.д., поки не буде завершено певний бізнес-процес. Тоді послідовність подій становить конкретну реалізацію бізнес-процесу.

Кожна подія описується з двох точок зору: інформаційної та процедурної. Інформаційно подія відбивається у вигляді деякого повідомлення, що фіксує факт виконання деякої функції зміни стану або появи нового. Процедурно подія викликає виконання нової функції, і тому для кожного стану об'єкта повинні бути задані опису цих викликів. Таким чином, події виступають в сполучній ролі для виконання функцій бізнес-процесів.

На зовнішньому рівні визначаються список зовнішніх подій, що викликаються взаємодією підприємства з зовнішнім середовищем (платежі податків, відсотків за кредитами, поставки за контрактами і т.д.), і список цільових установок, яким повинні відповідати бізнес-процеси (регламент виконання процесів, підтримка рівня матеріальних запасів, рівень якості продукції і т.д.).

На концептуальному рівні встановлюються бізнес-правила, що визначають умови виконання функцій при виникненні подій і досягненні станів об'єктів.

На внутрішньому рівні виконується формалізація бізнес-правил у вигляді тригерів або викликів програмних модулів.

Організаційна структура

Організаційна структура являє собою сукупність організаційних одиниць, як правило, пов'язаних ієрархічними і процесними відносинами. Організаційна одиниця - це підрозділ, що представляє собою об'єднання людей (персоналу) для виконання сукупності загальних функцій або бізнес-процесів. У функціонально-орієнтованої організаційної структури організаційна одиниця виконує набір функцій, що відносяться до однієї функції управління і входять до різні процеси. У процесно-орієнтованої структури організаційна одиниця виконує набір функцій, що входять в один тип процесу і відносяться до різних функцій управління.

На зовнішньому рівні будується структурна модель підприємства у вигляді ієрархії підпорядкування організаційних одиниць або списків взаємодіючих підрозділів.

На концептуальному рівні для кожного підрозділу задається організаційно-штатна структура посад (ролей персоналу).

На внутрішньому рівні визначаються вимоги до прав доступу персоналу до автоматизуються функцій інформаційної системи.

Технічна структура

Топологія визначає територіальне розміщення технічних засобів за структурними підрозділами підприємства, а комунікація - технічний спосіб реалізації взаємодії структурних підрозділів.

На зовнішньому рівні моделі визначаються типи технічних засобів обробки даних і їх розміщення по структурним підрозділам.

На концептуальному рівні визначаються способи комунікацій між технічними комплексами структурних підрозділів: фізичне переміщення документів, машинних носіїв, обмін інформацією по каналах зв'язку і т.д.

На внутрішньому рівні будується модель "клієнт-серверної" архітектури обчислювальної мережі.

Описані моделі предметної області націлені на проектування окремих компонентів ІС: даних, функціональних програмних модулів, керуючих програмних модулів, програмних модулів інтерфейсів користувачів, структури технічного комплексу. Для більш якісного проектування зазначених компонентів потрібна побудова моделей, що погоджує різні компоненти ІС між собою. У найпростішому випадку в якості таких моделей взаємодії можуть використовуватися матриці перехресних посилань: "функційні об'єкти", "функції-події", "організаційні одиниці - функції", "організаційні одиниці - об'єкти", "організаційні одиниці - технічні засоби" і т д . Такі матриці не наочні і не відображають особливості реалізації взаємодій.

Для правильного відображення взаємодій компонентів ІС важливо здійснювати спільне моделювання таких компонентів, особливо з змістовної точки зору об'єктів і функцій. Методологія структурного системного аналізу істотно допомагає в рішенні таких задач.

Структурним аналізом прийнято називати метод дослідження системи, який починається з її загального огляду, а потім деталізується, набуваючи ієрархічну структуру з дедалі більшим числом рівнів. Для таких методів характерно: розбиття на рівні абстракції з обмеженим числом елементів (від 3 до 7); обмежений контекст, що включає тільки істотні деталі кожного рівня; використання строгих формальних правил запису; послідовне наближення до результату. Структурний аналіз заснований на двох базових принципах - "розділяй і володарюй" і принципі ієрархічної впорядкованості. Рішення важких проблем шляхом їх розбиття на безліч менших незалежних завдань (так званих "чорних ящиків") і організація цих завдань в деревовидні ієрархічні структури значно підвищують розуміння складних систем. Визначимо ключові поняття структурного аналізу.

Операція - елементарне (неподільне) дія, що виконується на одному робочому місці.

Функція - сукупність операцій, згрупованих за певною ознакою.

Бізнес-процес - пов'язана сукупність функцій, в ході виконання якої споживаються певні ресурси і створюється продукт (предмет, послуга, наукове відкриття, ідея), що представляє цінність для споживача.

Подпроцесс - це бізнес-процес, який є структурним елементом деякого бізнес-процесу і представляє цінність для споживача.

Бізнес-модель - структуроване графічне опис мережі процесів і операцій, пов'язаних з даними, документами, організаційними одиницями та іншими об'єктами, що відображають існуючу або передбачувану діяльність підприємства.

Існують різні методології структурного моделювання предметної області, серед яких слід виділити функціонально-орієнтовані та об'єктно-орієнтовані методології.

Функціонально-орієнтовані та об'єктно-орієнтовані методології опису предметної області

Процес бізнес-моделювання може бути реалізований в рамках різних методик, що відрізняються насамперед своїм підходом до того, що являє собою моделируемая організація. У відповідності з різними уявленнями про організацію методики прийнято ділити на об'єктні і функціональні (структурні).

Об'єктні методики розглядають моделируемую організацію як набір взаємодіючих об'єктів - виробничих одиниць. Об'єкт визначається як відчутна реальність - предмет або явище, що мають чітко визначається поведінка. Метою застосування даної методики є виділення об'єктів, що становлять організацію, і розподіл між ними відповідальностей за виконувані дії.

Функціональні методики, найбільш відомою з яких є методика IDEF, розглядають організацію як набір функцій, що перетворює надходить потік інформації в вихідний потік. Процес перетворення інформації споживає певні ресурси. Основна відмінність від об'єктної методики полягає в чіткому відділенні функцій (методів обробки даних) від самих даних.

З точки зору бізнес-моделювання кожен з представлених підходів має свої переваги. Об'єктний підхід дозволяє побудувати більш стійку до змін систему, краще відповідає існуючим структурам організації. Функціональне моделювання добре показує себе в тих випадках, коли організаційна структура знаходиться в процесі зміни або взагалі слабо оформлена. Підхід від виконуваних функцій інтуїтивно краще розуміється виконавцями при отриманні від них інформації про їх поточну роботу.

Функціональна схема IDEF0

Методологію IDEF0 можна вважати наступним етапом розвитку добре відомого графічного мови опису функціональних систем SADT (Structured Analysis and Design Technique). Історично IDEF0 як стандарт був розроблений в 1981 році в рамках великої програми автоматизації промислових підприємств, яка носила позначення ICAM (Integrated Computer Aided Manufacturing). Сімейство стандартів IDEF успадкувало своє позначення від назви цієї програми (IDEF = Icam DEFinition), і остання його редакція була випущена в грудні 1993 року Національним Інститутом за Стандартами та Технологій США (NIST).

Метою методики є побудова функціональної схеми досліджуваної системи, яка описує всі необхідні процеси з точністю, достатньою для однозначного моделювання діяльності системи.

В основі методології лежать чотири основні поняття: функціональний блок, інтерфейсна дуга, декомпозиція, глосарій.

Функціональний блок (Activity Box) являє собою деяку конкретну функцію в рамках даної системи. За вимогами стандарту назва кожного функціонального блоку має бути сформульовано в глагольном способі (наприклад, "виробляти послуги"). На діаграмі функціональний блок зображується прямокутником (рис. 6.1). Кожна з чотирьох сторін функціонального блоку має своє певне значення (роль), при цьому:

верхня сторона має значення "Управління" (Control);

ліва сторона має значення "Вхід" (Input);

права сторона має значення "Вихід" (Output);

нижня сторона має значення "Механізм" (Mechanism).

Інтерфейсна дуга (Arrow) відображає елемент системи, який обробляється функціональним блоком або надає інший вплив на функцію, представлену даними функціональним блоком. Інтерфейсні дуги часто називають потоками або стрілками.

За допомогою інтерфейсних дуг відображають різні об'єкти, в тій чи іншій мірі визначають процеси, що відбуваються в системі. Такими об'єктами можуть бути елементи реального світу (деталі, вагони, співробітники і т.д.) або потоки даних і інформації (документи, дані, інструкції і т.д.).

Залежно від того, до якої з сторін функціонального блоку підходить дана інтерфейсна дуга, вона носить назву "входить", "вихідної" або "керуючої".

Необхідно відзначити, що будь-який функціональний блок за вимогами стандарту повинен мати, принаймні, одну керуючу интерфейсную дугу і одну вихідну. Це і зрозуміло - кожен процес повинен відбуватися за якимись правилами (що відображається керуючої дугою) і повинен видавати деякий результат (що виходить дуга), інакше його розгляд не має ніякого сенсу.

Обов'язкова наявність керуючих інтерфейсних дуг є одним з головних відмінностей стандарту IDEF0 від інших методологій класів DFD (Data Flow Diagram) і WFD (Work Flow Diagram).

Декомпозиція (Decomposition) є основним поняттям стандарту IDEF0. Принцип декомпозиції застосовується при розбитті складного процесу на складові його функції. При цьому рівень деталізації процесу визначається безпосередньо розробником моделі.

Декомпозиція дозволяє поступово і структуровано представляти модель системи у вигляді ієрархічної структури окремих діаграм, що робить її менш перевантаженою і легко засвоюваній.

Останнім з понять IDEF0 є глосарій (Glossary). Для кожного з елементів IDEF0 - діаграм, функціональних блоків, інтерфейсних дуг - існуючий стандарт має на увазі створення і підтримку набору відповідних визначень, ключових слів, оповідних викладів і т.д., які характеризують об'єкт, відображений даними елементом. Цей набір називається глосарієм і є описом суті даного елемента. Глосарій гармонійно доповнює наочний графічний мову, забезпечуючи діаграми необхідної додатковою інформацією.

Модель IDEF0 завжди починається з представлення системи як єдиного цілого - одного функціонального блоку з інтерфейсними дугами, що тягнуться за межі даної області. Така діаграма з одним функціональним блоком називається контекстної діаграмою.

У пояснювальному тексті до контекстної діаграмі повинна бути вказана мета (Purpose) побудови діаграми у вигляді короткого опису і зафіксована точка зору (Viewpoint).

Визначення та формалізація мети розробки IDEF0-моделі є вкрай важливим моментом. Фактично мета визначає відповідні області в досліджуваній системі, на яких необхідно фокусуватися в першу чергу.

Точка зору визначає основний напрямок розвитку моделі та рівень необхідної деталізації. Чітке фіксування точки зору дозволяє розвантажити модель, відмовившись від деталізації і дослідження окремих елементів, які не є необхідними, виходячи з обраної точки зору на систему. Правильний вибір точки зору істотно скорочує тимчасові витрати на побудову кінцевої моделі.

Виділення подпроцессов. У процесі декомпозиції функціональний блок, який в контекстної діаграмі відображає систему як єдине ціле, піддається деталізації на іншій діаграмі. Отримана діаграма другого рівня містить функціональні блоки, що відображають головні подфункции функціонального блоку контекстної діаграми, і називається дочірньої (Child Diagram) по відношенню до нього (кожен з функціональних блоків, що належать дочірній діаграмі, відповідно називається дочірнім блоком - Child Box). У свою чергу, функціональний блок - предок називається батьківським блоком по відношенню до дочірньої діаграмі (Parent Box), а діаграма, до якої він належить - батьківської діаграмою (Parent Diagram). Кожна з подфункций дочірньої діаграми може бути далі деталізована шляхом аналогічної декомпозиції відповідного їй функціонального блоку. У кожному разі декомпозиції функціонального блоку все інтерфейсні дуги, що входять у цей блок або виходять з нього, фіксуються на дочірньої діаграмі. Цим досягається структурна цілісність IDEF0-моделі.

Іноді окремі інтерфейсні дуги вищого рівня не має сенсу продовжувати розглядати на діаграмах нижнього рівня, або навпаки - окремі дуги нижньої відображати на діаграмах більш високих рівнів - це буде тільки перевантажувати діаграми і робити їх складними для сприйняття. Для вирішення подібних завдань в стандарті IDEF0 передбачено поняття тунелювання. Позначення "тунелю" (Arrow Tunnel) у вигляді двох круглих дужок навколо початку інтерфейсної дуги позначає, що ця дуга була успадкована від функціонального батьківського блоку і з'явилася (з "тунелю") тільки на цій діаграмі. У свою чергу, таке ж позначення навколо кінця (стрілки) інтерфейсної дуги в безпосередній близькості від блоку-приймача означає той факт, що в дочірньої по відношенню до цього блоку діаграмі ця дуга відображатися і розглядатися не буде. Найчастіше буває, що окремі об'єкти і відповідні їм інтерфейсні дуги не розглядаються на деяких проміжних рівнях ієрархії, - в такому разі вони спочатку "занурюються в тунель", а потім при необхідності "повертаються з тунелю".

Зазвичай IDEF0-моделі несуть в собі складну і концентровану інформацію, і для того, щоб обмежити їх перевантаженість і зробити легким для читання, в стандарті прийняті відповідні обмеження складності.

Рекомендується представляти на діаграмі від трьох до шести функціональних блоків, при цьому кількість відповідних до одного функціонального блоку (що виходять з одного функціонального блоку) інтерфейсних дуг передбачається не більше чотирьох.

Стандарт IDEF0 містить набір процедур, що дозволяють розробляти і погоджувати модель великою групою людей, що належать до різних областей діяльності, що моделюється. Зазвичай процес розробки є ітеративним і складається з наступних умовних етапів:

Створення моделі групою фахівців, що відносяться до різних сфер діяльності підприємства. Ця група в термінах IDEF0 називається авторами (Authors). Побудова первинної моделі є динамічним процесом, протягом якого автори опитують компетентних осіб про структуру різних процесів, створюючи моделі діяльності підрозділів. При цьому їх цікавлять відповіді на наступні питання:

Що надходить до підрозділу "на вході"?

Які функції і в якій послідовності виконуються в рамках підрозділу?

Хто є відповідальним за виконання кожної з функцій?

Чим керується виконавець при виконанні кожної з функцій?

Що є результатом роботи підрозділу (на виході)?

На основі наявних положень, документів і результатів опитувань створюється чернетку (Model Draft) моделі.

Поширення чернетки для розгляду, погоджень і коментарів. На цій стадії відбувається обговорення чернетки моделі з широким колом компетентних осіб (в термінах IDEF0 - читачів) на підприємстві. При цьому кожна з діаграм чорновий моделі письмово критикується і коментується, а потім передається автору. Автор, в свою чергу, також письмово погоджується з критикою або відкидає її з викладом логіки прийняття рішення і знову повертає відкоригований чернетку для подальшого розгляду. Цей цикл триває до тих пір, поки автори і читачі не прийдуть до єдиної думки.

Офіційне затвердження моделі. Затвердження узгодженої моделі відбувається керівником робочої групи в тому випадку, якщо у авторів моделі і читачів відсутні розбіжності з приводу її адекватності. Остаточна модель являє собою узгоджене уявлення про підприємство (системі) з заданої точки зору і для заданої мети.

Наочність графічного мови IDEF0 робить модель цілком читається і для осіб, які не брали участі в проекті її створення, а також ефективної для проведення показів і презентацій. Надалі на базі побудованої моделі можуть бути організовані нові проекти, націлені на виробництво змін в моделі

Функціональна схема потоків даних

Метою методики є побудова моделі даної системи у вигляді діаграми потоків даних (Data Flow Diagram - DFD), що забезпечує правильне опис виходів (відгуку системи в вигляді даних) при заданому впливі на вхід системи (подачі сигналів через зовнішні інтерфейси). Діаграми потоків даних є основним засобом моделювання функціональних вимог до проектованої системи.

При створенні діаграми потоків даних використовуються чотири основні поняття: потоки даних, процеси (роботи) перетворення вхідних потоків даних у вихідні, зовнішні сутності, накопичувачі даних (сховища).

Потоки даних є абстракціями, що використовуються для моделювання передачі інформації (або фізичних компонент) з однієї частини системи в іншу. Потоки на діаграмах зображуються іменованими стрілками, орієнтація яких вказує напрямок руху інформації.

Призначення процесу (роботи) полягає в продукуванні вихідних потоків із вхідних відповідно до дії, що задається ім'ям процесу. Ім'я процесу повинно містити дієслово в невизначеною формі з подальшим доповненням (наприклад, "отримати документи з відвантаження продукції"). Кожен процес має унікальний номер для посилань на нього всередині діаграми, який може використовуватися спільно з номером діаграми для отримання унікального індексу процесу у всій моделі.

Сховище (накопичувач) даних дозволяє на зазначених ділянках визначати дані, які будуть зберігатися в пам'яті між процесами. Фактично сховище представляє "зрізи" потоків даних у часі. Інформація, яку воно містить, може використовуватися в будь-який час після її отримання, при цьому дані можуть вибиратися в будь-якому порядку. Ім'я сховища має визначати його вміст і бути іменником.

Зовнішня сутність є матеріальним об'єктом поза контекстом системи, що є джерелом або приймачем системних даних. Її ім'я повинно містити іменник, наприклад, "склад товарів". Передбачається, що об'єкти, що представлені як зовнішні сутності, не повинні брати участь ні в якій обробці.

Крім основних елементів, до складу DFD входять словники даних і миниспецификации.

Словники даних є каталогами всіх елементів даних, присутніх в DFD, включаючи групові та індивідуальні потоки даних, сховища і процеси, а також всі їх атрибути.

Мініспеціфікаціі обробки - описують DFD-процеси нижнього рівня. Фактично миниспецификации представляють собою алгоритми опису завдань, які виконуються процесами: безліч всіх мініспеціфікаціі є повною специфікацією системи.

Процес побудови DFD починається зі створення так званої основної діаграми типу "зірка", на якій представлено моделюється і всі зовнішні сутності, з якими він взаємодіє. У разі складного основного процесу він відразу представляється у вигляді декомпозиції на ряд взаємодіючих процесів. Критеріями складності в даному випадку є: наявність великої кількості зовнішніх сутностей, багатофункціональність системи, її розподілений характер. Зовнішні сутності виділяються по відношенню до основного процесу. Для їх визначення необхідно виділити постачальників і споживачів основного процесу, тобто всі об'єкти, які взаємодіють з основним процесом. На цьому етапі опис взаємодії полягає у виборі дієслова, що дає уявлення про те, як зовнішня сутність використовує основний процес або використовується ним. Наприклад, основний процес - "облік звернень громадян", зовнішня сутність - "громадяни", опис взаємодії - "подає заяви і отримує відповіді". Цей етап є принципово важливим, оскільки саме він визначає межі системи, що моделюється.

Для всіх зовнішніх сутностей будується таблиця подій, що описує їх взаємодію з основним потоком. Таблиця подій включає в себе найменування зовнішньої сутності, подія, його тип (типовий для системи або винятковий, що реалізується за певних умов) і реакцію системи.

На наступному кроці відбувається декомпозиція основного процесу на набір взаємозв'язаних процесів, які обмінюються потоками даних. Самі потоки не конкретизуються, визначається лише характер взаємодії. Декомпозиція завершується, коли процес стає простим, тобто .:

процес має два-три вхідних і вихідних потоку;

процес може бути описаний у вигляді перетворення вхідних даних у вихідні;

процес може бути описаний у вигляді послідовного алгоритму.

Для простих процесів будується Мініспеціфікація - формальний опис алгоритму перетворення вхідних даних у вихідні.

Миниспецификация задовольняє наступним вимогам: для кожного процесу будується одна специфікація; специфікація однозначно визначає вхідні та вихідні потоки для даного процесу; специфікація не визначає спосіб перетворення вхідних потоків у вихідні; специфікація посилається на наявні елементи, не вводячи нові; специфікація по можливості використовує стандартні підходи і операції.

Після декомпозиції основного процесу для кожного подпроцесса будується аналогічна таблиця внутрішніх подій.

Наступним кроком після визначення повної таблиці подій виділяються потоки даних, якими обмінюються процеси і зовнішні сутності. Найпростіший спосіб їх виділення полягає в аналізі таблиць подій. Події перетворюються в потоки даних від ініціатора події до запитуваного процесу, а реакції - в зворотний потік подій. Після побудови вхідних і вихідних потоків аналогічним чином будуються внутрішні потоки. Для їх виділення для кожного з внутрішніх процесів виділяються постачальники і споживачі інформації. Якщо постачальник або споживач інформації представляє процес збереження або запиту інформації, то вводиться сховище даних, для якого цей процес є інтерфейсом.

Після побудови потоків даних діаграма повинна бути перевірена на повноту і несуперечність. Повнота діаграми забезпечується, якщо в системі немає "висів" процесів, що не використовуються в процесі перетворення вхідних потоків у вихідні. Несуперечливість системи забезпечується виконанням наборів формальних правил про можливі типи процесів: на діаграмі не може бути потоку, який зв'язує дві зовнішні сутності - це взаємодія видаляється з розгляду; жодна сутність не може безпосередньо отримувати або віддавати інформацію в сховище даних - сховище даних є пасивним елементом, керованим за допомогою інтерфейсного процесу; два сховища даних не можуть безпосередньо обмінюватися інформацією - ці сховища повинні бути об'єднані.

До переваг методики DFD відносяться:

можливість однозначно визначити зовнішні сутності, аналізуючи потоки інформації всередині і поза системою;

можливість проектування зверху вниз, що полегшує побудову моделі "як має бути";

наявність специфікацій процесів нижнього рівня, що дозволяє подолати логічну незавершеність функціональної моделі і побудувати повну функціональну специфікацію розроблюваної системи.

До недоліків моделі віднесемо: необхідність штучного введення керуючих процесів, оскільки дії, що управляють (потоки) і керуючі процеси з точки зору DFD нічим не відрізняються від звичайних; відсутність поняття часу, тобто відсутність аналізу часових проміжків при перетворенні даних (всі обмеження по часу повинні бути введені в специфікаціях процесів).

Об'єктно-орієнтована методика

Принципова відмінність між функціональним і об'єктним підходом полягає в способі декомпозиції системи. Об'єктно-орієнтований підхід використовує об'єктну декомпозицію, при цьому статична структура описується в термінах об'єктів і зв'язків між ними, а поведінка системи описується в термінах обміну повідомленнями між об'єктами. Метою методики є побудова бізнес-моделі суб'єкта господарювання, що дозволяє перейти від моделі сценаріїв використання до моделі, яка визначає окремі об'єкти, які беруть участь в реалізації бізнес-функцій.

Концептуальною основою об'єктно-орієнтованого підходу є об'єктна модель, яка будується з урахуванням наступних принципів:

абстрагування;

інкапсуляція;

модульність;

ієрархія;

типізація;

паралелізм;

стійкість.

Основними поняттями об'єктно-орієнтованого підходу є об'єкт і клас.

Об'єкт - предмет або явище, що має чітко певну поведінку і володіють станом, поведінкою і індивідуальністю. Структура і поведінку подібних об'єктів визначають загальний для них клас. Клас - це безліч об'єктів, пов'язаних спільністю структури і поведінки. Наступну групу важливих понять об'єктного підходу складають успадкування і поліморфізм. Поняття поліморфізм може бути інтерпретовано як здатність класу належати більш ніж одному типу. Спадкування означає побудову нових класів на основі існуючих з можливістю додавання або перевизначення даних і методів.

Важливим якістю об'єктного підходу є узгодженість моделей діяльності організації і моделей проектованої інформаційної системи від стадії формування вимог до стадії реалізації. За об'єктним моделям може бути простежено відображення реальних сутностей модельованої предметної області (організації) в об'єкти і класи інформаційної системи.

Більшість існуючих методів об'єктно-орієнтованого підходу включають мова моделювання і опис процесу моделювання. Процес - це опис кроків, які необхідно виконати при розробці проекту. В якості мови моделювання об'єктного підходу використовується уніфікована мова моделювання UML, який містить стандартний набір діаграм для моделювання.

Діаграма (Diagram) - це графічне представлення безлічі елементів. Найчастіше вона зображується у вигляді зв'язного графа з вершинами (сутностями) і ребрами (відносинами) і являє собою деяку проекцію системи.

Об'єктно-орієнтований підхід має наступні переваги:

Об'єктна декомпозиція дає можливість створювати моделі меншого розміру шляхом використання загальних механізмів, що забезпечують необхідну економію виражальних засобів. Використання об'єктного підходу істотно підвищує рівень уніфікації розробки і придатність для повторного використання, що веде до створення середовища розробки і переходу до складального створення моделей.

Об'єктна декомпозиція дозволяє уникнути створення складних моделей, так як вона передбачає еволюційний шлях розвитку моделі на базі відносно невеликих підсистем.

Об'єктна модель природна, оскільки орієнтована на людське сприйняття світу.

До недоліків об'єктно-орієнтованого підходу відносяться високі початкові витрати. Цей підхід не дає негайної віддачі. Ефект від його застосування позначається після розробки двох-трьох проектів і накопичення повторно використовуваних компонентів. Діаграми, що відображають специфіку об'єктного підходу, менш наочні.

Порівняння існуючих методик

У функціональних моделях (DFD-діаграмах потоків даних, SADT-діаграмах) головними структурними компонентами є функції (операції, дії, роботи), які на діаграмах зв'язуються між собою потоками об'єктів.

Безсумнівним достоїнством функціональних моделей є реалізація структурного підходу до проектування ІС за принципом "зверху-вниз", коли кожен функціональний блок може бути декомпозирован на безліч підфункцій і т.д., виконуючи, таким чином, модульне проектування ІС. Для функціональних моделей характерні процедурна строгість декомпозиції ІС і наочність представлення.

При функціональному підході об'єктні моделі даних у вигляді ER-діаграм "об'єкт - властивість - зв'язок" розробляються окремо. Для перевірки коректності моделювання предметної області між функціональними і об'єктними моделями встановлюються взаємно однозначні зв'язки.

Головний недолік функціональних моделей полягає в тому, що процеси і дані існують окремо один від одного - крім функціональної декомпозиції існує структура даних, яка перебуває на другому плані. Крім того, не ясні умови виконання процесів обробки інформації, які динамічно можуть змінюватися.

Перераховані недоліки функціональних моделей знімаються в об'єктно-орієнтованих моделях, в яких головним структуроутворюючих компонентом виступає клас об'єктів з набором функцій, які можуть звертатися до атрибутів цього класу.

Для класів об'єктів характерна ієрархія узагальнення, що дозволяє здійснювати спадкування не тільки атрибутів (властивостей) об'єктів від вищого класу об'єктів до нижчому класу, але і функцій (методів).

У разі спадкування функцій можна абстрагуватися від конкретної реалізації процедур (абстрактні типи даних), які відрізняються для певних підкласів ситуацій. Це дає можливість звертатися до подібних програмним модулям за загальними іменами (поліморфізм) і здійснювати повторне використання програмного коду при модифікації програмного забезпечення. Таким чином, адаптивність об'єктно-орієнтованих систем до зміни предметної області в порівнянні з функціональним підходом значно вище.

При об'єктно-орієнтованому підході змінюється і принцип проектування ІС. Спочатку виділяються класи об'єктів, а далі в залежності від можливих станів об'єктів (життєвого циклу об'єктів) визначаються методи обробки (функціональні процедури), що забезпечує найкращу реалізацію динамічної поведінки інформаційної системи.

Для об'єктно-орієнтованого підходу розроблені графічні методи моделювання предметної області, узагальнені в мові уніфікованого моделювання UML. Однак по наочності представлення моделі користувачеві-замовнику об'єктно-орієнтовані моделі явно поступаються функціональним моделям.

При виборі методики моделювання предметної області зазвичай в якості критерію виступає ступінь її динамічності. Для більш регламентованих завдань більше підходять функціональні моделі, для більш адаптивних бізнес-процесів (управління робочими потоками, реалізації динамічних запитів до інформаційних сховищ) - об'єктно-орієнтовані моделі. Однак в рамках однієї і тієї ж ІС для різних класів задач можуть вимагатися різні види моделей, що описують одну й ту ж проблемну область. В такому випадку повинні використовуватися комбіновані моделі предметної області.

синтетична методика

Як можна бачити з представленого огляду, кожна з розглянутих методик дозволяє вирішити задачу побудови формального опису робочих процедур досліджуваної системи. Всі методики дозволяють побудувати модель "як є" і "як повинно бути". З іншого боку, кожна з цих методик має істотні недоліки. Їх можна підсумувати таким чином: недоліки застосування окремої методики лежать не в області опису реальних процесів, а в неповноті методичного підходу.

Функціональні методики в цілому краще дають уявлення про існуючі функціях в організації, про методи їх реалізації, причому чим вище ступінь деталізації досліджуваного процесу, тим краще вони дають можливість окреслити систему. Під кращим описом в даному випадку розуміється найменша помилка при спробі за отриманою моделі передбачити поведінку реальної системи. На рівні окремих робочих процедур їх опис практично однозначно збігається з фактичною реалізацією в потоці робіт.

На рівні загального опису системи функціональні методики допускають значний ступінь сваволі у виборі загальних інтерфейсів системи, її механізмів і т.д., тобто у визначенні меж системи. Добре описати систему на цьому рівні дозволяє об'єктний підхід, заснований на понятті сценарію використання. Ключовим є поняття про сценарії використання як про сеанс взаємодії дійової особи з системою, в результаті якого дійова особа отримує щось, що має для нього цінність. Використання критерію цінності для користувача дає можливість відкинути не мають значення деталі потоків робіт і зосередитися на тих функціях системи, які виправдовують її існування. Однак і в цьому випадку завдання визначення меж системи, виділення зовнішніх користувачів є складною.

Технологія потоків даних, історично виникла першою, легко вирішує проблему кордонів системи, оскільки дозволяє за рахунок аналізу інформаційних потоків виділити зовнішні сутності і визначити основний внутрішній процес. Однак відсутність виділених керуючих процесів, потоків і подієвої орієнтованості не дозволяє запропонувати цю методику в якості єдиної.

Найкращим способом подолання недоліків розглянутих методик є формування синтетичної методики, що об'єднує різні етапи окремих методик. При цьому з кожної методики необхідно взяти частину методології, найбільш повно і формально викладену, і забезпечити можливість обміну результатами на різних етапах застосування синергетичної методики. У бізнес-моделюванні неявним чином йде формування подібної синергетичної методики.

Ідея синтетичної методики полягає в послідовному застосуванні функціонального і об'єктного підходу з урахуванням можливості реінжинірингу існуючої ситуації.

Розглянемо застосування синтетичної методики на прикладі розробки адміністративного регламенту.

При побудові адміністративних регламентів виділяються наступні стадії:

Визначення меж системи. На цій стадії за допомогою аналізу потоків даних виділяють зовнішні сутності і власне моделируемую систему.

Виділення сценаріїв використання системи. На цій стадії за допомогою критерію корисності будують для кожної зовнішньої сутності набір сценаріїв використання системи.

Додавання системних сценаріїв використання. На цій стадії визначають сценарії, необхідні для реалізації цілей системи, відмінних від цілей користувачів.

Побудова діаграми активностей за сценаріями використання. На цій стадії будують набір дій системи, що призводять до реалізації сценаріїв використання;

Функціональна декомпозиція діаграм активностей як контекстних діаграм методики IDEF0.

Формальний опис окремих функціональних активностей у вигляді адміністративного регламенту (із застосуванням різних нотацій).