Міністерство освіти і науки України

Національний авіаційний університет

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра інженерії програмного забезпечення

**Домашнє завдання**

«Вибір моделі специфікації вимог в задачі архітектурного проектування ІС з використанням відомих методологій»

**Виконано студентом групи:**

ПІ-322, Царука С.О.

**Перевірено:**

Поставна Л.П.

Київ – 2021

**Мета лабораторної роботи – з**астосування набутих знань передпроектного обстеження прикладних систем для вирішення задачі узгодження моделі специфікації вимог до ІС з відомими методологіями архітектурного проектування ІС.

**Завдання**

1. Здійснити пошук інформаційних джерел з описами відомих поширених методологій архітектурного проєктування.

**Варіант 1**

1. Обрати методологію, яка в найбільшій мірі відповідає контексту архітектурного проектування ІС відповідно до індивідуального варіанту ПрС.
2. Здійснити реферативний опис обраної методології.
3. Провести оцінку відповідності вихідних даних для архітектурного проектування за обраною методологією розробленій в ЛР1-4 моделі ІТ-потреб ПрС.
4. Розробити пропозиції щодо складу, форми і змісту специфікацій моделі ІТ-потреб, які в найбільшій мірі відповідають обраній методології. При наявності у вибраній методології патернів архітектурного проектування обрати відповідний патерн і налаштувати його на умови виконання завдання.

**Тема**

Варіант: 30. Інформаційна система міської агенції нерухомості.

**Хід роботи**

***4.1. Реферативний опис обраної методології архітектурного проектування.***

ООП — методологія, основана на представленні програми у вигляді сукупності об’єктів, кожний з яких є об’єктом визначеного класу, а класи створюють ієрархію наслідування.

В даному визначені є три основні частини:

* в якості базових операцій використовуються об’єкти, а не алгоритми;
* кожний об’єкт є екземпляром якогось визначеного класу;
* класи організовані ієрархічно.

ООП дозволяє розкласти проблему на окремі задачі. Кожна задача стає самостійним об’єктом, який має свої дані і коди. Таким чином, задача спрощується і програміст отримує можливість оперувати набагато більшими по об’єму програмами. В центрі — об’єкт — осяжна сутність, яка виявляє свою поведінку.

Об'єктно-орієнтований метод використовує об'єктну декомпозицію. При цьому структура системи описується в термінах об'єктів і зв'язків між ними, а поведінка системи описується в термінах обміну повідомленнями між об'єктами.

Головне достоїнство об'єктно-орієнтованого підходу (ООП) можна сформулювати наступним способом: об'єктно-орієнтовані системи більш відкриті і легше піддаються внесенню змін, оскільки їх конструкція базується на стійких формах. Це дає можливість системі розвиватися поступово і не призводить до повної її переробки навіть у разі істотних змін вихідних вимог.

ООП має ряд переваг:

* Об'єктна декомпозиція дає можливість створювати програмні системи меншого розміру шляхом використання загальних механізмів, що забезпечують необхідну економію виражальних засобів. Використання ООП істотно підвищує рівень уніфікації розробки і придатність для повторного використання не тільки програмного забезпечення, але і проектів, що, врешті-решт, веде до складного створення. Системи часто виходять більш компактними, ніж їх не об'єктно-орієнтовані еквіваленти, що означає не тільки зменшення обсягу програмного коду, але і здешевлення проекту за рахунок використання попередніх розробок.
* Об'єктна декомпозиція зменшує ризик створення складних програмних систем, так як вона передбачає еволюційний шлях розвитку системи на базі відносно невеликих підсистем. Процес інтеграції системи розтягується на весь час розробки, а не перетворюється в одноразова подія.
* Об'єктна модель цілком природна, оскільки в першу чергу орієнтована на людське сприйняття світу, а не на комп'ютерну реалізацію.
* Об'єктна модель дозволяє в повній мірі використовувати виражальні можливості об'єктних і об'єктно-орієнтованих мов програмування.

***4.2. Опис результатів порівняльного аналізу контексту архітектурного проектування ІС ПрС з профілем кращого застосування методології проектування.***

Архітектура відповідає вимогам для проектування ІС відповідно до обраної ПрС. Архітектурні процеси поділяються на 3 рівня: рівень подання, прикладний рівень та рівень керування ресурсами. Таким чином, можна використати об’єктно-орієнтований підхід, створивши різні рівні роботи інформаційної системи. Розподіл процесів буде наступним:

Процеси рівня подання (представлення):

* Процес пошуку послуг – процес для подання запиту від користувача до системи;
* Процес перегляду даних послуги – процес, що надає детальну інформацію про послугу на запит користувача;
* Процес замовлення/підтвердження послуги – процес для замовлення/підтвердження послуги для користувача;
* Процес перегляду замовлень – процес для відображення та перегляду детальних даних замовлень користувача.

Процеси прикладного рівня (бізнес-логіки):

* Процес оновлення системи – процес для внесення поправок до роботи системи;
* Процес отримання даних – процес для обміну даними між рівнями системи;
* Процес редагування даних – процес для зміни даних;
* Процес супроводу системи – процес для перевірки системи, виправлення пошкоджень, відстеження стану;
* Процес створення замовлення – процес, що отримує дані від користувача та створює на їх основі замовлення.

Процеси рівня керування ресурсами (обробки даних):

* Процес підбору даних – процес для знаходження необхідних даних, що відповідають запиту;
* Процес збереження даних – процес збереження нових/змінених даних.

***4.3. Пропозиції щодо моделі специфікації вимог архітектурного проектування, яка в найбільшій мірі відповідає контексту задачі проектування для відповідної ПрС.***

Специфікація вимог. На даному етапі аналітик ретельно досліджує та безжалісно переформульовує вимоги, конструюючи моделі, виходячи з концепцій системи. Він повинен тісно працювати із замовником, щоб досягти розуміння завдання, тому що визначення, отримані на попередньому етапі, рідко виявляються повними або коректними.

Пропонується використовувати аналітичну модель, оскільки вона концентрується на описі об’єктів, що повністю відповідає концепції ООП.

Аналітична модель – це стисла й точна абстракція того, що саме повинна робити система. Аналітична модель не повинна містити ніяких рішень щодо реалізації.

Аналітична модель складається з двох частин: моделі предметної області (domain model) – опису об’єктів реального світу, що відображає система, та моделі програмного додатка (application model) – опису видимих користувачеві частин самого додатка. Модель предметної області, у свою чергу, складається з моделі класів та моделі взаємодії. Об’єкти моделі додатка можуть керувати здійсненням торгів і відображати результати. Гарна модель повинна бути доступною для розуміння та критики з боку експертів, які не є програмістами.

***4.4. Опис (при наявності) патернів, фреймворків, які доцільно застосувати в процесі архітектурного проектування ІС ПрС за обраною методологією.***

Основний патерн архітектури: багаторівнева система.

Відповідно до патерна "Багаторівнева система" структурні елементи системи організуються в окремі рівні з взаємопов'язаними обов'язками таким чином, щоб на нижньому рівні розташовувалися низькорівневі служби та служби загального призначення, а на більш високих - об'єкти рівня логіки додатка. При цьому взаємодія і зв'язування рівнів відбувається зверху вниз. Зв'язування об'єктів знизу вгору слід уникати.

За даного патерну шар подання охоплює все, що має відношення до спілкування користувача з системою. До основних функцій шару подання відноситься відображення інформації й інтерпретація користувачем команд з перетворенням їх у відповідні операції в контексті домену (бізнес - логіка) і джерела даних. Джерело даних - підмножина функцій, що забезпечує взаємодію зі сторонніми системами, які виконуються.

Запропоновано використовувати наступні фреймворки:

* Фреймворк Захмана;
* Фреймворк TOGAF.

Фреймворк Захмана є одним із самих старих архітектурних фреймворків. Він був створений співробітником компанії ІBM Джоном Захманом (John Zachman). Захман заклав в основу свого фреймворку класифікацію (таксономію) артефактів системи. Серед них можна виділити дані, функціональність, моделі, специфікації й документи. У результаті, можна вважати цей фреймворк онтологією верхнього рівня, що описує конкретну систему.

Для побудови таксономії Захманом запропоновано відповісти на шість питань про функціонування організації: що, як, де, хто, чому. Дані питання ставляться до наступних аспектів системи:

* використовувані дані (що?);
* процеси й функції (як?);
* місця виконання процесів (де?);
* організації й персоналії (хто?);
* керуючої події (коли?);
* мети й обмеження, що визначають роботу системи (чому?).

Відповідати на ці питання необхідно з різним ступенем деталізації. Описано шість рівнів:

* рівень контексту;
* рівень бізнесів-описів;
* системний рівень;
* технологічний рівень;
* технічний рівень;
* рівень реальної системи.

Фреймворк TOGAF (The Open GroupArchіtecture Framework) являє собою набір засобів для розробки архітектур різного призначення. З його допомогою інформаційна система подається як сукупність модулів.

В рамках TOGAF дається особливе визначення архітектури – «формальний опис системи, або детальний план системи на рівні компонентів і методології їх реалізації». Загальноприйняте ж визначення архітектури (відповідно до стандарту ANSI / IEEE 1471-2000) визначається як «опис організації системи в термінах компонентів, їх взаємозв’язків між собою і з навколишнім середовищем і принципи управління їх розробкою і розвитком».

TOGAF складається з чотирьох архітектурних доменів:

* бізнес-архітектура (описує ключові бізнес-процеси, стратегію розвитку бізнесу і принципи управління);
* архітектура рівня додатків (описує інтерфейси додатків і способи їх застосування в термінах бізнес-сервісів);
* архітектура рівня даних (визначає логічну й фізичну структуру даних в організації);
* технологічна архітектура (визначає програмну, апаратну і мережеву інфраструктури).

Слід зазначити, що TOGAF може використовуватися не тільки як єдиний фреймворк при розробці, але і в сукупності з іншими фреймворками.

Головними складовими частинами TOGAF є:

* ADM-методика (Architecture Development Method), що описує процес розробки архітектури;
* керівництва і методики проектування для ADM;
* фреймворк архітектурного опису (Architecture Content Framework), що є детально відпрацьованої моделлю результатів розробки;
* архітектурний континуум організації (Enterprise Continuum), у вигляді репозиторію архітектурних артефактів і реалізацій;
* еталонні моделі TOGAF (TOGAF Reference Models): TRM (Technical Reference Model) – технічна еталонна модель; III-RM (The Integrated Information Infrastructure Model) – інтегрована модель інформаційної інфраструктури.
* фреймворк, що описує структуру організації, її персонал, необхідні ролі і рівні відповідальності (Architecture Capability Framework).

Відповідно до методики ADM архітектурний процес можна розбити на дев’ять фаз. ADM представляє з себе ітераційний процес, який відбувається на двох рівнях. На верхньому рівні кожної ітерації повторюються загальні для кожної з фаз дії. Нижній рівень описує ітерації всередині кожної фази. Рішення приймаються на підставі існуючих вимог бізнесу та існуючих рішень.

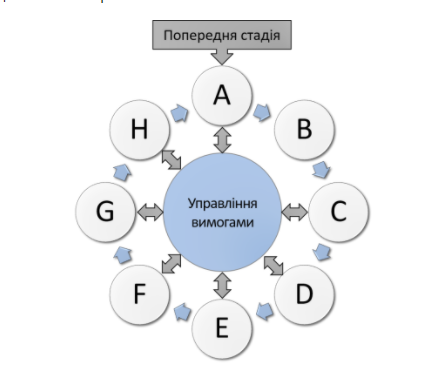


Рис.2. Схема фаз відповідно до метрики ADM.

Зміст фаз архітектурного процесу методики ADM:

|  |  |
| --- | --- |
| Стадія процесу | Опис |
| Попередня фаза (Preliminary Phase) | Визначення способів управління, меж розробки, принципів реалізації, уточнення моделі щодо специфіки . |
| Фаза A, розробка загального подання (Architecture Vision) | Створення загального уявлення про архітектуру, визначення меж проекту, затвердження плану робіт і завдань для виконавців |
| Фаза B, розробка бізнес-архітектури (Business Architecture) | Виявлення принципів функціонування організації, принципів управління проектом |
| Фаза C, розробка інформаційної архітектури (Information Systems Architecture) | Розробка архітектури даних і додатків |
| Фаза D, розробка технологічної архітектури (Technology  Architecture) | Підбір апаратних засобів, засобів мережевої інфраструктури, механизмів їх взаємодії |
| Фаза E, Можливості і рішення (Opportunities and Solutions) | Реалізація розробленої архітектури (купити готове рішення або зробити власне) |
| Фаза F, Планування переходу до нової архітектури (Migration Planning) | Оцінка ризиків, аналіз деталей переходу |
| Фаза G, формування системи керування реалізацією (Implementation Governance) | Моніторинг процесу впровадження і специфікація виникаючих проблем |
| Фаза H, керування зміною архітектури (Architecture Change Management) | Налагодження процесу керування змінами розробленої архітектури |

**Висновки**

Під час виконання лабораторної роботи було **з**астосувано набуті знання передпроектного обстеження прикладних систем для вирішення задачі узгодження моделі специфікації вимог до ІС з відомими методологіями архітектурного проектування ІС.