

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

# СЦЕНАРНИЙ АНАЛІЗ БІЗНЕС ПРОЦЕСІВ

## Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського  
як навчальний посібник для здобувачів ступеня бакалавра  
за освітньою програмою «Інженерія програмного забезпечення інтелектуальних  
кібер-фізичних систем в енергетиці»  
спеціальності 121 Інженерія програмного забезпечення

Укладачі: В. О. Кузьмініх, О. В. Коваль, Р. А. Тараненко

Електронне мережеве навчальне видання

Київ  
КПІ ім. ІГОРЯ СІКОРСЬКОГО  
2024

УДК 005.8 (075.8)

К89

Укладачі:

Кузьмініх Валерій Олександрович, канд. техн. наук, доц.  
Коваль Олександр Васильович, д-р техн. наук, проф.  
Тараненко Руслан Анатолійович

Рецензент

*Тарнавський Ю. А., к.ф.-м.н., доц., кафедра цифрових технологій в енергетиці, ННІ АТЕ, «КПІ ім. Ігоря Сікорського»*

Відповідальний

редактор

*Гаврилко Є. В., д.т.н., проф., заступник директора з навчально-виховної роботи ННІ АТЕ, «КПІ ім. Ігоря Сікорського»*

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського  
(протокол № 5 від 29.01.2024 р.)  
за поданням вченої ради ННІ АТЕ  
(протокол № 7 від 29.01.2024 р.)*

### **Кузьмініх В. О.**

К89

Сценарний аналіз бізнес процесів [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спец. 121 Інженерія програмного забезпечення / В. О. Кузьмініх, О. В. Коваль, Р. А. Тараненко ; КПІ ім. Ігоря Сікорського. Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2024. – 189 с.

Навчальний посібник «Сценарний аналіз бізнес-процесів» створено для фахівців сфери інформаційних технологій та систем і призначено для оволодіння основами процесного підходу та сценарного аналізу бізнес-процесів. Матеріал викладено у обсязі необхідному в підготовці здобувачів ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». Посібник призначено для забезпечення проведення лекційних занять. Подано загальний та теоретичний матеріал, розглянуто основні поняття, методології, стандарти та нотації, аналіз та моделювання. Okремо розглянуто застосування мікросервісів у аналітичній діяльності.

УДК 005.8 (075.8)

Реєстр. № НП 23/24-312. Обсяг 9,31 авт. арк.

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
проспект Берестейський, 37, м. Київ, 03056  
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© КПІ ім. Ігоря Сікорського, 2024

2024

# ЗМІСТ

<b>ЗМІСТ</b>	3
<b>ВСТУП</b>	5
<b>ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА АКРОНІМІВ</b>	6
<b>РОЗДІЛ 1. БІЗНЕС-ПРОЦЕСИ. ОСНОВНІ ПОНЯТТЯ</b>	8
<b>Питання до розділу 1</b>	15
<b>РОЗДІЛ 2. СИСТЕМИ БІЗНЕС-ПРОЦЕСІВ</b>	16
<b>Питання до розділу 2</b>	21
<b>РОЗДІЛ 3. БІЗНЕС-ПРОЦЕСИ ТА СЦЕНАРІЇ ДІЙ</b>	22
<b>Питання до розділу 3</b>	27
<b>РОЗДІЛ 4. МЕТОДОЛОГІЇ ОПИСУ БІЗНЕС-ПРОЦЕСІВ</b>	28
<b>Питання до розділу 4</b>	34
<b>РОЗДІЛ 5. НОТАЦІЇ ОПИСУ БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ</b>	35
<b>Питання до розділу 5</b>	46
<b>РОЗДІЛ 6. ОПИС БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ З ВИКОРИСТАННЯМ     БЛОК-СХЕМ ПРОЦЕСІВ.</b>	48
<b>Питання до розділу 6</b>	55
<b>РОЗДІЛ 7. МЕТОДОЛОГІЯ SADT. НОТАЦІЯ IDEF0</b>	56
<b>Питання до розділу 7</b>	68
<b>РОЗДІЛ 8. ОПИС ЛОГІКИ СЦЕНАРІЇВ У МЕТОДОЛОГІЇ IDEF3</b>	69
<b>Питання до розділу 8</b>	83
<b>РОЗДІЛ 9. ВИКОРИСТАННЯ CASE-ТЕХНОЛОГІЇ В МОДЕЛЮВАННІ БІЗНЕС-     ПРОЦЕСІВ ТА СЦЕНАРІЇВ</b>	84
<b>Питання до розділу 9</b>	92
<b>РОЗДІЛ 10. ОБ'ЄКТНО-ОРІЄНТОВАНИЙ АНАЛІЗ І ПРОЕКТУВАННЯ</b>	93
<b>Питання до розділу 10</b>	99
<b>РОЗДІЛ 11. МОДЕЛЮВАННЯ СТАТИЧНОЇ СТРУКТУРИ СИСТЕМИ</b>	100
<b>Питання до розділу 11</b>	118
<b>РОЗДІЛ 12. ПОСЛІДОВНІСТЬ ВИКОНАННЯ ДІЙ БІЗНЕС-ПРОЦЕСІВ ТА     СЦЕНАРІЇВ</b>	119

<b>Питання до розділу 12</b>	127
<b>РОЗДІЛ 13. ПРОЦЕСНИЙ АНАЛІЗ У ПОБУДОВІ СЦЕНАРІЇВ</b>	128
<b>Питання до розділу 13</b>	135
<b>РОЗДІЛ 14. АНАЛІЗ БІЗНЕС-ДІЯЛЬНОСТІ</b>	136
<b>Питання до розділу 14</b>	146
<b>РОЗДІЛ 15. МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ ІНФОРМАЦІЙНИХ СИСТЕМ</b>	147
<b>Питання до розділу 15</b>	154
<b>РОЗДІЛ 16. ПОМИЛКИ У ОПИСІ БІЗНЕС-ПРОЦЕСІВ СКЛАДНИХ ВИРОБНИЧИХ ОБ'ЄКТІВ ІНФОРМАТИЗАЦІЇ</b>	155
<b>Питання до розділу 16</b>	162
<b>РОЗДІЛ 17. МЕТОДИКА ФОРМУВАННЯ СЦЕНАРІЮ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ</b>	163
<b>Питання до розділу 17</b>	175
<b>РОЗДІЛ 18. ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ В РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ</b>	176
<b>Питання до розділу 18</b>	184
<b>ПЕРЕЛІК ПОСИЛАНЬ</b>	185
<b>СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ</b>	187

## ВСТУП

Інтенсивний розвиток та впровадження автоматизованих систем управління підприємствами та організаціями, багатофункціональних інформаційно-аналітичних систем, систем пошуку та обробки великих даних, які базуються на необхідності виконання великої кількості послідовних дій по вилученню, збереженню та обробці даних, приводить до необхідності побудови моделей відповідних бізнес-процесів та сценаріїв окремих дій.

Процесний підхід в управлінській діяльності почав формуватися досить давно, але свій реальний розвиток отримав лише у другій половині ХХ століття у зв'язку з активним розвитком інформаційних технологій та програмних систем, які почали активно створюватись для інформатизації та автоматизації діяльності підприємств та організацій.

Сучасна аналітична діяльність, як частина управлінської діяльності, теж тісно пов'язана з обробкою великих обсягів інформації. Це визначає необхідність побудови відповідних моделей бізнес-процесів та сценаріїв збору, обробки та аналізу даних, для підвищення ефективності аналітичної діяльності. Одним з ефективних методів, що дозволяє значно прискорити побудову моделей та покращити ефективність аналітичної діяльності є сценарний підхід.

У посібнику розглянуто методологію побудови системи управління організаціями на підставі процесного та функціонально-ієрархічного підходу, особливості побудови інформаційно-аналітичних процесів, показано способи моделювання, характерні особливості і можливості окремих нотацій опису бізнес-процесів та сценаріїв послідовності дій.

Навчальний посібник містить ознайомчий курс «Сценарний аналіз бізнес-процесів», що орієнтований на розробників програмних систем. В посібнику розглянуто основні поняття, методології, стандарти та нотації, що використовуються для моделювання та аналізу різноманітних бізнес-процесів та сценаріїв. Окремо розглянуто застосування мікросервісної архітектури у розробці програмних засобів управління сценаріями в аналітичній діяльності.

Матеріал навчального посібника викладено у обсязі необхідному в підготовці здобувачів ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». Посібник призначено для забезпечення проведення лекційних занять.

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА АКРОНІМІВ

<b>АМД</b>	Аналіз міжнародної діяльності
<b>АРМ</b>	Автоматизоване робоче місце
<b>АТС</b>	Автоматична телефонна станція
<b>АС</b>	Автоматизована система
<b>БП</b>	Бізнес процес
<b>ВПС</b>	Військово-повітряні сили
<b>ЖЦ</b>	Життєвий цикл
<b>ІАС</b>	Інформаційно-аналітична система
<b>ІС</b>	Інформаційна система
<b>КІАС</b>	Корпоративна інформаційно аналітична система
<b>КІС</b>	Корпоративна інформаційна система
<b>ЛП</b>	Лінгвістичний процесор
<b>МО</b>	Мікрооперації
<b>ОА</b>	Об'єкт автоматизації
<b>ООП</b>	Об'єктно-орієнтоване програмування
<b>ПБ</b>	Прізвище Ім'я по Батькові
<b>ПІН</b>	Персональний ідентифікаційний номер
<b>САД</b>	Сценарій аналітичної діяльності
<b>ЦФВ</b>	Центр фінансової відповідальності
<b>ARIS</b>	Architecture of Integrated Information Systems - методологія і програмний продукт компанії IDS Scheer для моделювання бізнес-процесів компанії
<b>BPM</b>	Business Process Management - управління бізнес-процесами
<b>BPМN</b>	Business Process Model and Notation - нотація і модель бізнес-процесів
<b>BPML</b>	Business Process Modeling Language - мова моделювання бізнес-процесів
<b>BPMS</b>	Business Process Management System – система управління бізнес-процесами
<b>BPR</b>	Business Process Reengineering - реінжиніринг бізнес-процесів
<b>BS</b>	Business studio - програмний продукт для моделювання бізнесу
<b>CASE</b>	Computer Aided Software Engineering – комп'ютерне автоматизоване програмне забезпечення

<b>DFD</b>	Data Flow Diagram – діаграма потоків даних
<b>eEPC</b>	Extended Event-driven Process Chain - розширена нотація опису ланцюжка процесу, керованого подіями.
<b>ELMA</b>	Компанія, що займається системами управління бізнес-процесами
<b>ERD</b>	Entity-Relationship Diagrams - діаграма «сутність-зв'язок» (ER-модель)
<b>ERP</b>	Enterprise Resource Planning - Система планування ресурсів підприємства
<b>ICAM (I-CAM)</b>	Integrated Computer-Aided Manufacturing – інтегроване комп'ютерне автоматизоване виробництво
<b>ICOM</b>	Абревіатура: Input - Введення, Control - Управління, Output - Висновок, Mechanism – Механізм
<b>IDEF</b>	I-CAM DEFinition або Integrated DEFinition — методології сімейства ICAM
<b>ISO</b>	International Organization for Standardization - Міжнародна організація з стандартизації
<b>IT</b>	Information Technology – інформаційні технології
<b>MS</b>	Microsoft - Майкрософт
<b>OOAD</b>	Object-Oriented Analysis/Design - Об'єктно-орієнтований аналіз і проектування (ООАП)
<b>OSTN</b>	Object State Transition Network - діаграми переходу стану об'єкта
<b>PFDD</b>	Process Flow Description Diagrams - визначення діаграми потоку процесів
<b>PCD</b>	Process Chain Diagram – діаграма ланцюжка процесів
<b>SADT</b>	Structured Analysis and Design Technique - методологія структурного аналізу та проектування
<b>SLD</b>	Swim Lane Diagram - діаграма «Плавальні доріжки»
<b>TQM</b>	Total quality management - загальне управління якістю
<b>UML</b>	Unified Modeling Language – уніфікована мова моделювання
<b>UOB</b>	Unit of Behavior – одиниця поведінки
<b>UOW</b>	Unit of Work – одиниця роботи
<b>VAD</b>	Value added chain diagram - діаграми ланцюжка доданої цінності

# РОЗДІЛ 1. БІЗНЕС-ПРОЦЕСИ. ОСНОВНІ ПОНЯТТЯ

## 1.1. Визначення поняття бізнес-процес

Визначення поняття «бізнес-процес» [1, 4, 5, 6, 10,]:

1. Бізнес-процес (БП) – це сукупність взаємопов'язаних заходів, дій або робіт, спрямованих на створення певного продукту або послуги для споживачів.
2. Бізнес-процес – керована, цілеспрямована послідовність взаємопов'язаних дій, що повторюються за визначеною процедурою, з урахуванням існуючих обмежень на використання ресурсів та стану оточуючого середовища. Бізнес-процес перетворює входи у виходи, які цінні для користувача/споживача цього бізнес-процесу.
3. Бізнес-процес – це відносно постійний цикл взаємопов'язаних операцій, що безпосередньо стосуються здійснення господарської діяльності, виконуваних для досягнення певної фінансової вигоди.

## 1.2. Основні складові бізнес-процесу

Основні складові елементи БП (рис. 1.1, 1.2):

1. Будь-який бізнес-процес має вхід, вихід, управління, ресурси і обмеження.
2. Вхід – це матеріальні та/або інформаційні об'єкти і дії, що ініціюють бізнес-процес та використовуються бізнес-процесом для отримання результату (виходу). Допускається, що бізнес-процес може не мати входу.
3. Управління – це правила, посадові інструкції, технології, протоколи, процедури або стандарти, якими керується бізнес-процес.
4. Вихід – матеріальні та/або інформаційні об'єкти і дії, що є результатом виконання бізнес-процесу. Бізнес-процес без результату не має сенсу та значення.
5. Ресурси – фінанси, персонал підприємства, устаткування, інструменти тощо.
6. Обмеження завжди є на вхід, вихід, управління, регламенти.

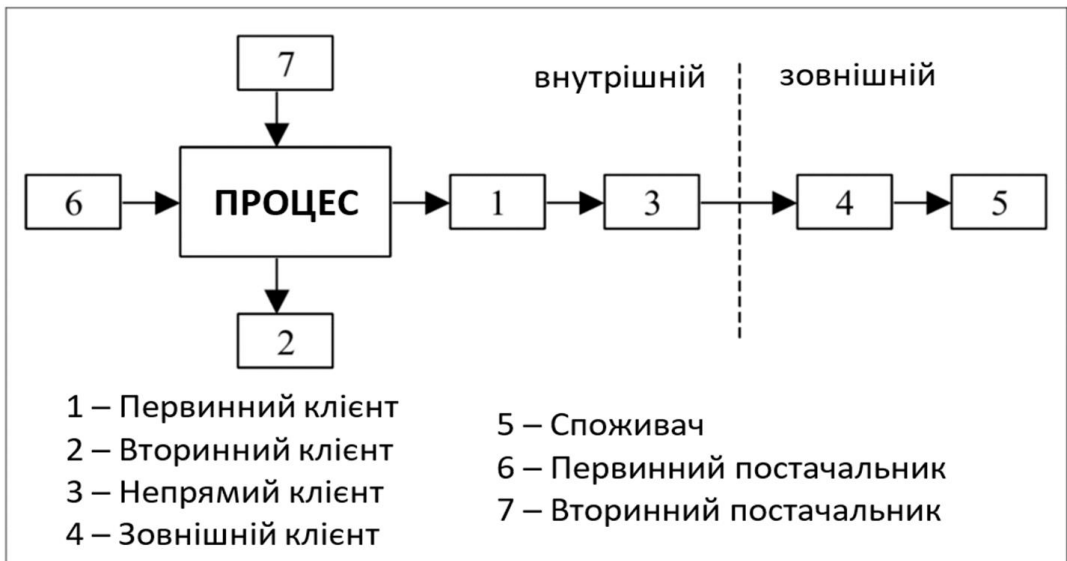


Рис. 1.1 Входи та виходи бізнес - процесу

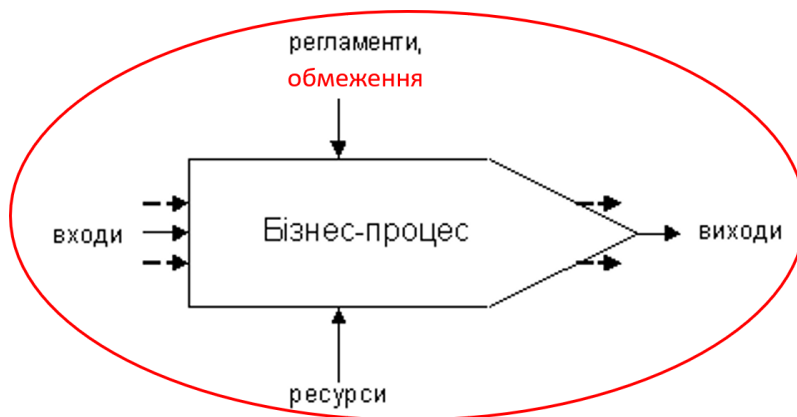


Рис. 1.2 Схема бізнес-процесу

Наприклад:

- входи – вимоги споживача до послуги;
- виходи – послуга, що відповідає вимогам споживача;
- ресурси – трудові, фінансові, матеріальні ресурси, обладнання;
- регламенти – нормативно-правова, технологічна документація, регламент БП;
- обмеження – на входи, на виходи (не перевищення вартості послуги для сприйняття споживачем; технічні обмеження, що впливають зі стану розвитку технологій тощо), на реалізацію БП (обмеження фінансових ресурсів, обмеження в часі, відсутність технічних рішень тощо).

### 1.3. Опис бізнес-процесів

Модель бізнес-процесу не може бути побудована без опису бізнес-процесу (таблиця 1.1, 1.2), який достатньо повно та адекватно відображає основні елементи структури бізнес-процесу як цілісної системи.

Опис нерідко є суб'єктивним, тому описуючи навіть «те, що є» (as is), завжди виникають певні складності, недоліки, невідповідності та відсутність повноти опису.

При необхідності описати «те, що має бути» (to be) розроблено, доводиться створювати щось нове на існуючій базі, як програмній, так і технічній.

Таблиця 1.1 Приклади табличного представлення процесів

	Крок	Початок	Відповідальний	Дія №1	Дія №2	Результат
1	Дзвінок клієнта	Вхідний дзвінок	Оператор контакт-центру	Пошук клієнта за номером телефону або П.І.Б.	Реєстрація нової картки (якщо нема запису)	Ідентифікація клієнта або реєстрація нової картки
2	Перевірка наявності товару	Клієнтське замовлення	Оператор контакт-центру Співробітник складського приміщення	перевірка наявності по базі даних, уточнення у робітника складу	Озвучування наявності, поточної ціни, способів та строків доставки	Перехід до презентації товару чи до очікування його надходження
3	Презентація товару	Замовлення клієнта + товар в наявності	Оператор контакт-центру Технічний фахівець	Проведення презентації, озвучування головних характеристик, відповіді на питання клієнта	Переведення на продаж, переклад на повернення, відмова від продажу	Перехід на етап оформлення чи завершення бізнес-процесу
4	Оформлення замовлення	Підтвердження замовлення від клієнта	Оператор контакт-центру Співробітник складу	Резерв товару на складі, оформлення заявки на доставку, підготовка документації	Сповіщення клієнта про приблизний чи точний час доставки	Закінчення оформлення та передача в службу доставки

Не варто забувати про можливість майбутнього поліпшення. Навіть, при бажанні створити бездоганний продукт, будуть виявлені корективи, які можна ввести в будь-який момент.

Автоматизація БП потребує значних інвестицій. Тому варто завжди враховувати реальні можливості змін as is -- > to be.

Таблиця 1.2 Приклад схеми бізнес-процесів банку

Об'єкт управління	Бізнес-процес
Бізнес-система	Аналізу та контролю діяльності
Технологія	Розробка нових і модифікація наявних продуктів
Клієнт	Продаж активних продуктів
Постачальники фінансових ресурсів	Залучення ресурсів
Персонал	Підготовка персоналу
Технологічні засоби	Підтримка технологічних засобів
Фінанси і стійкість банку	Управління фінансами
Операційна діяльність	Управління операційною діяльністю

#### 1.4. Класифікація бізнес-процесів

Типи процесів:

1. Основні – спрямовані на надання товарів або послуг, які є важливими об'єктами для компанії та відповідають за отримання прибутку.
2. Супутні – створюють побічні товари чи послуги, які служать наслідком основної підприємницької діяльності і забезпечують заробіток.
3. Допоміжні – застосовуються для реалізації головних процесів і підтримки їх специфічних рис.
4. Забезпечуючі – необхідні для нормального функціонування інших процесів та націлені на підтримку їх універсальних характеристик.

5. Управління – одночасно охоплюючи кілька керуючих функцій на всіх рівнях. Це такі процеси, як планування, аналіз стану, моніторинг робіт.

6. Розвитку – це процеси поліпшення створюваного товару або послуги, поліпшення обладнання та оснащення.

Таблиця. 1.3 Властивості 3 основних параметрів БП

Тривалість	Вартість	Задоволеність
Чим вище швидкість виконання процесу, тим краще продуктивність підприємства. Але важливо пам'ятати, що якість результату має відповідати високому показнику з меншими тимчасовими витратами. Для зниження часу виконання процесу застосовуються різні технічні та ІТ-методи, що прискорюють бізнес.	Вартість виконання бізнес-процесу необхідно зменшувати до мінімуму. Така вимога відноситься як до виробничого процесу, так і до надання послуг. Фірма, що оптимізує та знижує ціну, має значно більший заробіток.	Результатом бізнес-процесу є продукт, який максимально задовольняє споживача та замовника. Від якості фінального результату залежить успіх підприємства і лояльність клієнтської бази. Потрібно підсумовувати досвід у зворотному зв'язку для розкриття контролю якості.

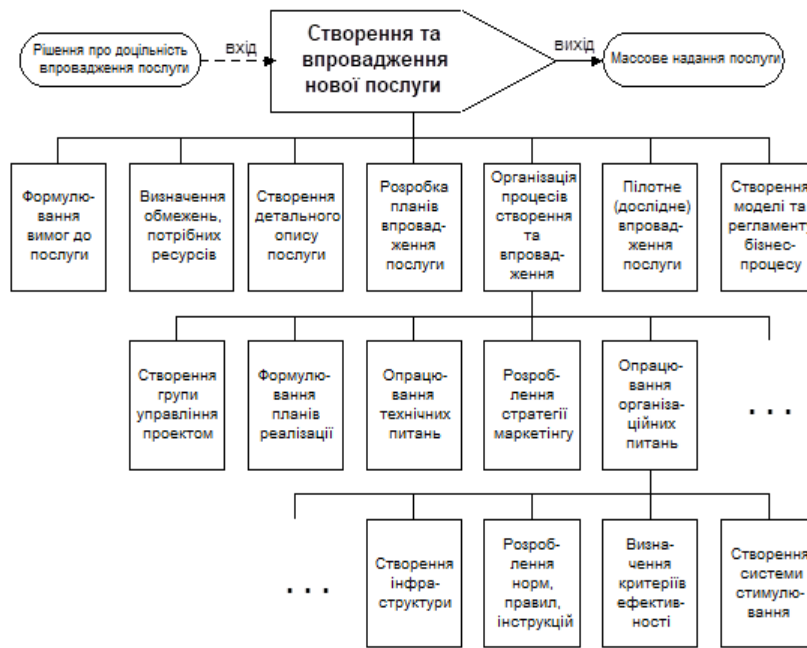


Рис. 1.4 Приклад послідовної декомпозиції

Виділяють 3 основні параметри бізнес-процесів (таблиця 1.3):

- тривалість;
- вартість;
- задоволеність споживача товаром або послугою.

### 1.5. Ієрархічне представлення бізнес-процесів. Декомпозиція

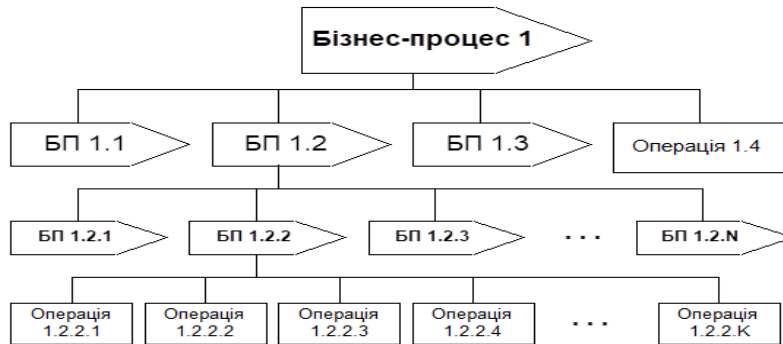


Рис. 1.5 Ієрархічне представлення бізнес-процесу

Більшість бізнес-процесів при певному рівні розгляду мають ієрархічну структуру(рис. 1.5).

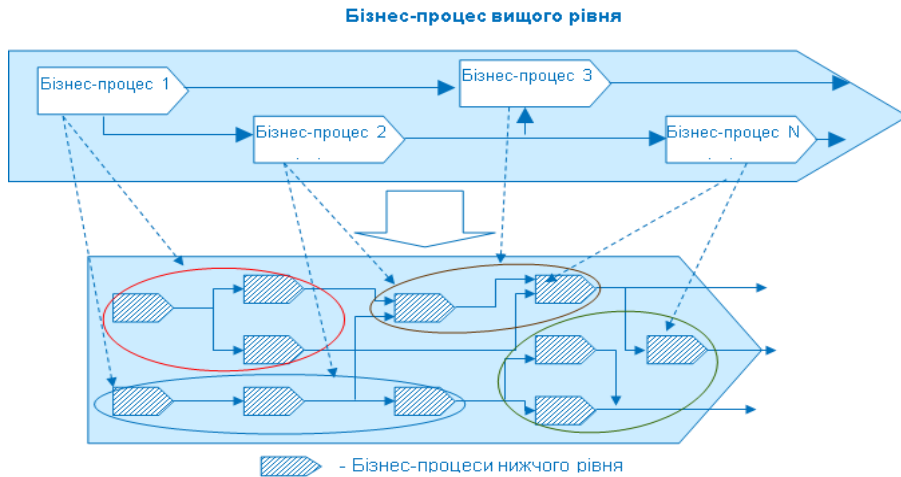


Рис. 1.6 Входження БП нижчого рівня у БП вищого рівня

Деталізація бізнес-процесів до рівня його підпроцесів називається декомпозицією (рис 1.4).

Декомпозиція – це прийом, що дозволяє представити складну систему у вигляді декількох більш простих взаємопов'язаних, вкладених систем.

Деталізація БП до рівня його підпроцесів називається декомпозицією першого рівня БП. Можна декомпонувати БП до будь-якого рівня залежно від потреби (рис. 1.6).

Така форма подання дозволяє аналізувати процес, не перевантажуючи уявлення елементами, зайвими для вирішення поточної задачі.

Глибина декомпозиції визначається цілями моделювання і, таким чином, визначає рівень детальності опису процесу. За аналогією з плануванням можна проводити моделювання та опис бізнес-процесів «зверху - вниз» і «знизу - вгору».

Мета декомпозиції дуже проста – якщо великим процесом складно управляти, його необхідно розділити на частини. Проектування бізнес процесів дозволяє нам «розбирати» і «збирати» процеси, змінюючи їх розмір.

Найнижчим рівнем деталізації БП є операції, що здійснюються певними виконавцями згідно з визначеними ролями відповідно до задокументованих інструкцій.

## **1.6. Власник бізнес-процесів**

Власник бізнес-процесу це особа, яка несе повну відповідальність за:

- актуальність регламенту БП та його відповідність вимогам зовнішнього середовища;
- здійснення вибіркового контролю виконання екземплярів БП;
- якість запланованих результатів виконання БП;
- постійний моніторинг БП;
- узгодження суттєвих умов з власниками суміжних БП

Власник БП має право:

- розпоряджатися ресурсами згідно з регламентом;
- отримувати у повному обсязі інформацію щодо виконання екземплярів БП;
- ініціювати (надавати керівництву пропозиції щодо) внесення змін до БП.

Власник БП не несе відповідальності за реалізацію кожного екземпляра БП.

## **1.7. Характерні риси бізнес-процесів. Формалізація для автоматизації**

Бізнес-процес має такі характерні риси:

- це цілеспрямована діяльність, що спрямована на перетворення входу БП у вихід;
- це поточна діяльність, вона постійно повторюється. При цьому бізнес-процес залишається незмінним, а його екземпляри можуть відрізнятись залежно від кожної конкретної комбінації входів та обмежень при його повтореннях;
  - входи та виходи БП мають бути визначені (специфіковані);
  - в ефективно функціонуючій системі БП повинен бути цінним для користувача цього БП. БП не має сенсу, якщо його вихід ніким не використовується, тобто не поступає на вхід іншого бізнес-процесу (внутрішнього або зовнішнього);
  - діяльність з перетворення входу БП у вихід має бути регламентована;
  - умови реалізації БП визначаються станом зовнішнього оточення БП.

Бізнес-процес вважається визначеним (формалізованим) для автоматизації, якщо для нього встановлені та задокументовані:

- входи;
- виходи;
- межі;
- обмежуючі фактори, умови реалізації;
- процедури виконання операцій;
- необхідні ресурси (матеріальні, трудові, інші);
- матриця відповідальності;
- правила управління бізнес-процесом;
- процедура моніторингу.

## **Питання до розділу 1**

1. Як визначається поняття «бізнес-процес»?
2. Чим характерні входи та виходи бізнес-процесу?
3. Які є типи бізнес-процесів?
4. Які є основні властивості бізнес-процесів?
5. Що таке декомпозиція бізнес-процесів?
6. Які характерні риси бізнес-процесів?

## РОЗДІЛ 2. СИСТЕМИ БІЗНЕС-ПРОЦЕСІВ

### 2.1. Характеристики системи

Взаємопов'язаність це риса притаманна системам.

Центральним поняттям системного аналізу БП є поняття "система" [1, 4, 5, 6, 10,].

Система є сукупність елементів (підсистем).

При певних умовах елементи самі можуть розглядатися як системи, а досліджувана система – як елемент більш складної системи:

- зв'язки між елементами в системі БП перевершують, по силі зв'язку, зв'язки цих елементів з елементами, що не входять в систему. Це властивість дозволяє виділити систему з середовища;
- для будь-якої системи БП характерно існування інтеграційних якостей (властивість емерджентності від англ. emergence - виникнення, поява нового), які притаманні системі в цілому, але не властиві жодному її елементу окремо;
- систему БП не можна зводити до простої сукупності елементів, не пов'язаних особливими, системно утворюючими зв'язками; незвідність властивостей системи до суми властивостей її компонентів (елементів «системний ефект»);
- системі притаманні мета та цілі (цільові функції) існування, для яких вона функціонує.

В даний час однозначного, чіткого визначення складної системи немає. Так до складних систем часто відносять системи, що мають 50-100 елементів; до надскладних - системи, що складаються з 100-1000 елементів; і до супер систем - системи з 1000 та більше елементів.

Англійська кібернетик С. Бір пропонує до складних відносити системи, що описують мовою теоретико-імовірнісних методів.

Визначення складної та простої системи:

1. Складною системою називається система, в моделі якої недостатньо інформації для ефективного управління цією системою.
2. Таким чином, ознакою простоти системи є достатність інформації для її управління. Якщо ж результат управління, отриманий за допомогою моделі, буде невідповідним, то таку систему відносять до складної.
3. Для розгляду системи в якості простої необхідне отримання достатньої інформації про неї і включення її в модель.

Від складних систем необхідно відрізнити великі системи.

Система, для актуалізації моделі якої в цілях управління бракує матеріальних ресурсів (машинного часу, ємності пам'яті, інших матеріальних засобів моделювання), називається великою.

До таких систем відносяться:

- економічні;
- організаційно-управлінські;
- нейрофізіологічні;
- біологічні системи та інші.

Способом переходу великих систем в прості є створення нових більш потужних засобів обчислювальної техніки.

Поняття великої і складної системи є різними. Однак в літературі ці поняття визначені неоднозначно.

Створення реальної системи БП означає, що вона синтезується з деяких компонентів в наступному порядку:

1. задум системи БП;
2. аналіз БП;
3. виділення компонентів БП;
4. моделювання БП;
5. розробка компонент БП;
6. об'єднання компонентів в єдине ціле;
7. верифікація (від лат. *verus* - "істинний" і *facere* - «робити»).

Спостерігач – це особа, яка представляє об'єкт або процес у вигляді системи. Від мети і завдань спостерігача (замовника, користувача ІС) залежить те, як буде розглядатися і як буде описана система.

Під елементом прийнято розуміти найпростішу неподільну частину системи.

У загальному вигляді існує множина таких частин, а спосіб їх виділення залежить від формулювання цілей аналізу системи.

Якщо в якості елемента системи прийняті поняття, що пов'язані між собою певними відношеннями, то маємо справу з символічними (абстрактними) системами. Прикладом таких систем служать алгоритмічні мови, системи числення, алгоритми.

Різальні (речові, фізичні) системи включають в себе щонайменше два фізичних об'єкта.

Система може бути розчленована на елементи не відразу, а шляхом послідовного поділу на підсистеми. Підсистеми самі є системами і до них, отже, відноситься все, що сказано про систему, в тому числі і про її цілісності.

Цим підсистема відрізняється від простої сукупності елементів, не об'єднаних метою і властивістю цілісності.

Система може бути представлена простим перерахуванням елементів, або завданням властивості приналежності до деякої множини, або послідовним розчленуванням на підсистеми, компоненти, елементи з взаємозв'язками між ними.

Поняття "структура" зазвичай відображає найбільш суттєві взаємозв'язки між елементами і їх групами.

Дані взаємозв'язки забезпечують існування системи та її основні властивості. Структурні властивості мають відносну незалежність від елементів і можуть виступати як інваріант при переході від однієї системи до іншої, що переносять закономірності, виявлені в одній з них, на іншу (навіть якщо ці системи мають різну фізичну природу).

## **2.2. Основні риси бізнес-процесу як системи**

Структура БП може бути представлена графічним відображенням, теоретико-множинним ставленням, у вигляді матриць. Вид уявлення системи залежить від мети відображення.

Зв'язки:

- визначають властивості системи БП;
- характеризують структуру і будову БП;
- визначають функціонування БП.

Зв'язки характеризуються:

- напрямком (спрямовані - неспрямовані; прямі і зворотні);
- силою (слабкі - сильні);
- характером (зв'язки підпорядкування, зв'язки породження, зв'язки рівноправності, зв'язки управління та інші).

Передбачається, що зв'язки існують між усіма системними елементами і підсистемами.

Функції системи БП – це діяльність, робота, зовнішній прояв властивостей якого-небудь об'єкта в даній системі БП. Функції класифікуються за різними ознаками залежно від цілей дослідження (щось нас цікавить, а щось ні).

Часто під функціями розуміється призначення і можливості системи.

Властивості системи БП. Це якості параметрів системи, тобто зовнішні прояви того способу, за допомогою якого отримують знання про систему. Властивості дають можливість описувати об'єкти системи кількісно, виражаючи їх у одиницях, що мають

певну розмірність. При цьому вони можуть змінюватися в результаті функціонування системи.

Стан – це миттєва характеристика (зупинка в розвитку) системи, яка забезпечує визначення знання властивостей системи в конкретний момент часу.

Стан визначається через вхідні дії і вихідні результати.

Поточний стан – в загальному випадку функціонально визначається входом, попереднім станом і виходом та часом визначення для конкретного екземпляру БП.

Поведінка – це зміна стану системи БП, результатом якого є певний результат. В основному термін "поведінка" відносять до людино-машинних або організаційних систем. Для технічних систем зазвичай говорять про процеси в системі.

Рівновага – це поняття визначається, як здатність системи за відсутності зовнішніх впливів зберігати свій стан невизначено тривалий час.

Під стійкістю розуміється здатність системи повертатися в стан рівноваги після впливу зовнішніх збурень.

Стан рівноваги, це стан в який система здатна повертатися після змін у надзвичайних, нерегламентованих ситуаціях (наприклад – БП управління транспортом під час тривоги). Для технічних систем поняття стійкості може бути визначено строго. Для людино-машинних і організаційних систем це поняття в значній мірі визначається якісно. Стан рівноваги є характерною ознакою для програмних смарт систем.

Під розвитком будемо розуміти послідовну зміну станів системи від деякого зафіксованого моменту часу. Характер цих змін визначається процесами, що йдуть в системі та взаємодією з навколишнім середовищем. Зміни можуть бути монотонними, стрибкоподібними, з повторенням вже пройдених станів (циклічний розвиток system upgrade).

Мета – це ідеальний результат діяльності в майбутньому. Вона визначає те, заради чого існує система БП.

Мета – це одне з ключових понять системного аналізу, що лежить в основі розвитку системи і забезпечує її цілеспрямованість (доцільність).

Мету можна визначити як бажаний результат діяльності, досяжний в межах деякого інтервалу часу Мета стає завданням, що стоїть перед системою, якщо зазначений термін її досягнення має конкретизовані кількісні характеристики бажаного результату.

Мета досягається в результаті вирішення задачі або ряду завдань, якщо вихідна мета може бути піддана поділу на деяку сукупність простіших (приватних) підзадач.

Цілі (цільові функції) – набір можливостей, що забезпечує система БП.

### 2.3. Ознаки системи бізнес-процесів

Основні ознаки системи БП:

1. Цілісність, зв'язність та відносна незалежність від середовища і інших систем. Зі зникненням пов'язаності зникає і система, хоча елементи системи і навіть деякі відносини між ними можуть бути збережені.

2. Наявність підсистем та зв'язків між ними або наявність структури системи. Зі зникненням підсистем або зв'язків між ними може зникнути і сама система.

3. Можливість відокремлення або абстрагування від навколишнього середовища, тобто відносна відокремленість від тих факторів середовища, які в достатній мірі не впливають на досягнення мети.

4. Зв'язки з навколишнім середовищем (комунікативність) з обміну ресурсами та інформацією.

5. Підпорядкованість всієї сукупності елементів системи певній меті.

6. Емерджентність або незвідність властивостей системи до властивостей елементів.

7. Наявність в ряді випадків змагального моменту, тобто необхідно враховувати конкуренцію окремих частин при функціонуванні системи.

8. Багатоаспектність проявів системи (технічна, економічна, соціальна, психологічна).

9. Контр інтуїтивність (причина і наслідок жорстко однозначно не пов'язані ні в часі, ні в просторі).

10. Нерегулярне, стохастично розподілене в часі надходження зовнішній впливів.

11. Ієрархічність. Під ієрархією розуміється послідовна декомпозиція вихідної системи на ряд рівнів з встановленням відносини підпорядкованості нижчих рівнів тим, що розміщені вище.

Сукупність взаємопов'язаних та взаємодіючих БП, що охоплюють всі основні операції (функції), які виконуються в підрозділах підприємства організації чи компанії, складає систему бізнес-процесів. Таким чином, системою бізнес-процесів описується вся їх діяльність. С точки зору автоматизації підприємства організації чи компанії вони можуть розглядатись як об'єкт автоматизації (ОА).

Метою формування системи БП є аналіз та оптимізація бізнес-моделі ОА для координації всіх складових системи, що дає можливість бачити взаємозв'язок та взаємозалежність бізнес-процесів в рамках цього ОА.

## **Питання до розділу 2**

1. Як визначається поняття «система» бізнес-процесів?
2. Як створюється системи бізнес-процесів?
3. Які основні поняття системи?
4. Як характеризуються зв'язки у системі бізнес-процесів?
5. Які є характеристики бізнес-процесів як системи?
6. Яка мета побудови системи бізнес-процесів?
7. Які характерні ознаки системи бізнес-процесів?

## РОЗДІЛ 3. БІЗНЕС-ПРОЦЕСИ ТА СЦЕНАРІЇ ДІЙ

### 3.1. Основні визначення поняття «сценарій»

Взагалі існує безліч різних визначень поняття «сценарій». Одні дослідники розуміють під сценарієм якусь статичну картинку майбутнього, інші – послідовність подій у часі; також виділяють безліч відмінностей і за іншими характеристиками сценарію [1, 2, 4, 5, 6, 7, 10, 12, 14, 16]. Іноді виділяють загальні характеристики сценаріїв, це:

- гіпотетична природа сценарію;
- опис альтернативних варіантів майбутнього;
- ланцюжок подій;
- причинно-наслідковий зв'язок;
- наявність внутрішньої узгодженості;
- основу чи базис для дій.

В результаті узагальнення цих характеристик приймемо в якості найбільш загального визначення сценарію наступне:

Сценарій – це послідовний опис альтернативних гіпотетично можливих варіантів розвитку подій в майбутньому, який відображає різні точки зору на минуле, сьогодення і майбутнє, а також який може бути базисом для планування дій.

Успішне застосування сценарного підходу низкою різноманітних компаній, а також зростання його популярності в державному стратегічному управлінні надало йому значної популярності та розповсюдженості. Однією з найважливіших рис сценарного підходу є його універсальність, гнучкість, можливість застосовувати різноманітні методи на різних етапах сценарного дослідження та управління процесами. Розробка сценарного підходу з використанням ряду аналітичних методів сприятиме підвищенню достовірності та обґрунтованості отриманих результатів. Це дає збільшення ефективності його застосування в стратегічному управлінні складними інформаційно-аналітичними системами.

На сьогодні існує багато методів вибору необхідних за якістю та ефективністю сценаріїв з множини можливих чи допустимих сценаріїв, які побудовані на основі аналізу складу та характеру факторів, що впливають на процес планування сценарію. В практиці планування виділяються ситуації, що відрізняються числом факторів, на основі яких приймають рішення та визначають принципові відмінності процедур формування необхідних сценаріїв. Однією з досить складних та актуальних на сьогодні задач, що потребують побудови ефективних сценаріїв для її вирішення, є

задача побудови та подальшої оптимізації сценаріїв для збору та аналізу різноманітних процесів діяльності організацій. Задача оптимізації сценаріїв збору та аналізу інформації на мережі є однією з найважливіших задач обробки великих об'ємів даних (big data).

Що є основою появи сценарного підходу та сценаріїв :

1. Посадові інструкції виконавців для робіт, які мають бути автоматизовані у рамках створення програмного забезпечення.

2. Протоколи (регламенти) виконання функцій по окремих задачах та окремими виконавцями(дуже поширено у США).

3. Визначені бізнес-процеси в організаціях з розподілом:

– по виконавцям;

– по окремим задачам, які виконуються як частина загального бізнес-процесу;

– по виконавцям для окремих задач.

Особливості та відмінності бізнес-процесів та сценаріїв:

– бізнес-процеси визначають склад, структуру та архітектуру програмного забезпечення, що розроблюється;

– сценарії дій визначають функції виконавців, їх повноваження, їх ступінь допуску до окремих функцій системи та можливості впливу на інформацію (додавання, виключення, зміни, корегування, розповсюдження та таке інше), що є в наявності в програмній системі, та визначають інтерфейси автоматизованих робочих місць користувачів програмної системи.

### **3.2. Формування рольових сценаріїв**

Розглядаючи методи побудови бізнес-процесів на основі використання описів у вигляді блок-схем у нотації Swim Lane, було визначено можливість побудови рольових сценаріїв.

Цей тип сценаріїв визначає такі послідовності дій кожної з ролей, які характерні для цих ролей, як частина бізнес-процесу що відповідає цій ролі.

Як правило перелік цих дій визначено посадовими інструкціями, регламентами, стандартами чи протоколами. Такий підхід характерний для державних організацій різного рівня та великих і структурованих приватних підприємств незалежно від напрямку їх діяльності.

Формування низки сценаріїв дає можливість не тільки чітко показати деталі бізнес-процесів, які мають бути інформатизовані, але точно визначити особливості і можливості кожного автоматизованого робочого місця (АРМу) для кожної з ролей.

Це дає можливість на етапі написання коду розробити програмні застосунки відповідні до потреб користувачів інформаційної системи.

### 3.3. Зв'язок структурних посадових інструкцій, регламентів бізнес процесів та рольових сценаріїв

У більшості більш-менш значних організацій та підприємств незалежно від форми їх організації та підпорядкування існують певні посадові інструкції, які визначають повноваження та можливі дії кожного з виконавців, які можуть приймати участь як учасники різних ролей у бізнес-процесах організації чи підприємства. Крім того додатком до цих посадових інструкцій можуть бути робочі інструкції, що визначають дії виконавців як окремі ролі у бізнес-процесах організації.

У ряді країн прийнято використання такого поняття як протокол дій, що у повній мірі відповідає поняттю сценарію дій у певних умовах та обставинах. Сукупність таких протоколів фактично відповідає посадовим повноваженням певних посадовців, що, в свою чергу, формує сукупність бізнес-процесів, в яких приймає участь цей посадовець.

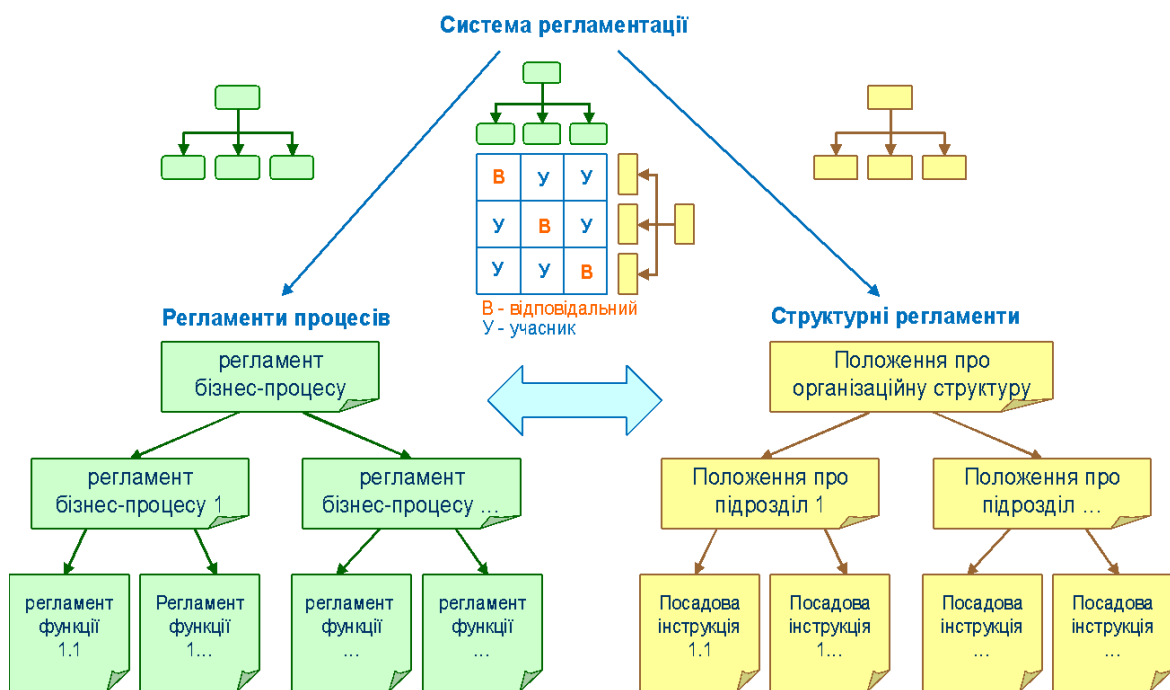


Рис. 3.1 Зв'язок структурних посадових інструкцій та регламентів БП

Єдина система регламентації (рис. 3.1) у рамках посадових повноважень, що визначені певними посадовими інструкціями, та регламентів бізнес-процесів дозволяє узгодити можливі протиріччя у описах дій та більш точно визначити всі необхідні елементи посадових інструкцій та повноважень посадовців відповідно до їх компетенцій у визначених бізнес-процесах.

Зв'язок між структурними регламентами у вигляді посадових інструкцій та регламентами процесів у вигляді сценаріїв дій відповідних бізнес-процесів дає можливість побудувати систему регламентацій з метою повного, всебічного та відповідного опису виконання функцій виконавців у дійсних бізнес-процесах, що визначають наявну діяльність підприємства чи організації.

### **3.4. Особливості формування рольових сценаріїв**

Формування рольових сценаріїв широко використовується при розробці програмних засобів з значною кількістю користувачів, коли дуже важливо визначити відповідальних та виконавців певних функцій та межі їх дій.

Такий підхід дає можливість:

- точно визначити рамки функцій, що реалізується програмним застосунком (score-аналіз) кожної з ролей;
- визначити інформаційні ресурси (БД та інше) необхідні для виконання функцій кожною з ролей у рамках конкретного бізнес-процесу;
- визначити функції програмного інтерфейсу для кожної з ролей для відповідної реалізації автоматизованого робочого місця (АРМу) кожної з ролей;
- визначити зв'язки та залежності у виконанні дій з іншими ролями та їх діями у рамках конкретного бізнес-процесу.

При формуванні рольових сценаріїв широко використовується підхід заснований на використанні описів бізнес-процесів у вигляді Swim Lane Diagrams (рис. 3.2).

Це дає можливість у описі одного бізнес-процесу відобразити всі сценарії дій для всіх ролей, що приймають участь у цьому бізнес-процесі.

Підхід на основі Swim Lane Diagrams дуже поширений у використанні при описі бізнес-процесів нижнього рівня, коли треба чітко визначити виконавців конкретних дій у кожному з бізнес-процесів.

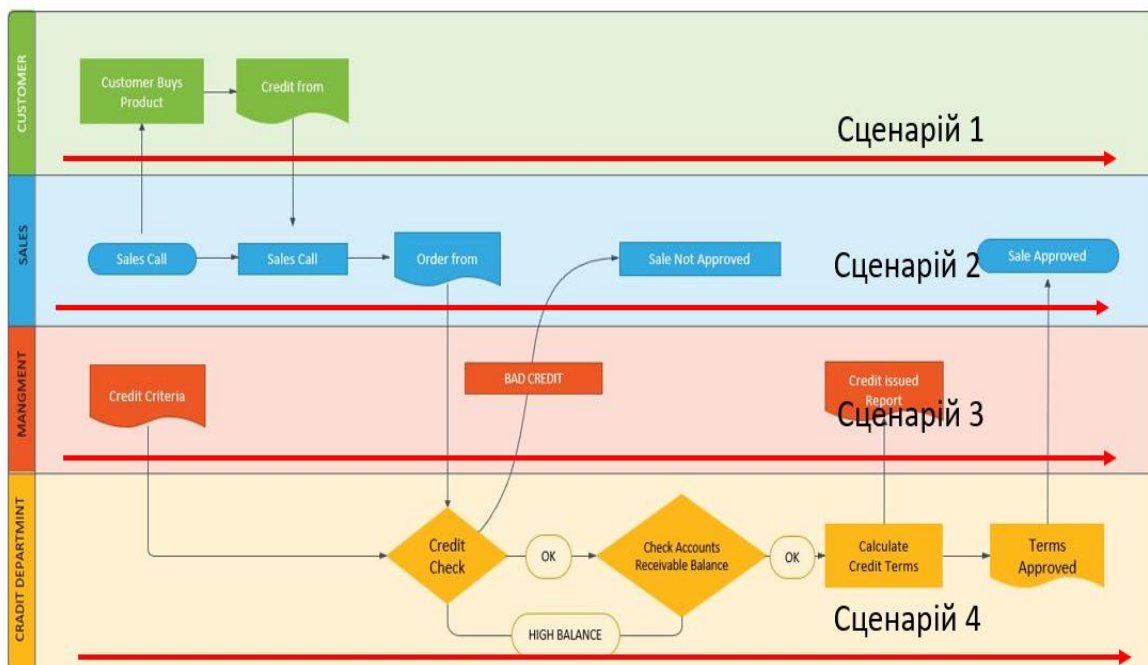


Рис. 3.2 Використання Swim Lane Diagrams у формуванні ролевих сценаріїв

Як показано на рис. 3.2 нотація Swim Lane Diagrams дозволяє на одному графіку бізнес-процесу відобразити чотири сценарії дій відповідно до їх виконавців (чи відповідальних). Це значно спрощує як виділення окремих сценаріїв у бізнес-процесах, так і встановлення належних зв'язків між окремими сценаріями у рамках єдиного загального бізнес-процесу.

### 3.5. Формування сценаріїв послідовності дій

Сценарії послідовності дій дають можливість чітко визначити усі можливі ситуації, що виникають під час реалізації кожного з бізнес-процесу.

Крім того такий підхід до опису сценаріїв цього типу дає можливість визначити межі (score) можливих чи допустимих послідовностей дій у процесі та напрямків розвитку можливих екземплярів процесу, що підлягає інформатизації. Це особливо важливо у описів процесів, що мають технічні чи інформаційні обмеження по доступним та дозволеним станам процесів що моделюються (з великим ступенем відповідальності за результати чи обмеженістю інформації про хід їх проходження).

### 3.6. Особливості формування сценаріїв послідовності дій

Побудова сценаріїв послідовності дій особливо ефективна для використання, коли треба точно визначити можливі та допустимі ланцюжки дій в залежності від ситуацій, що виникають у рамках дії бізнес-процесу.

Це особливо важливо для складних процесів, моделі яких мають ієрархічну структуру, та які мають декілька шляхів розвитку процесу в залежності від ситуацій, що виникають під час розвитку процесу у конкретному екземплярі процесу.

Моделі таких бізнес-процесів описуються з використанням зворотних шляхів та можуть мати декілька альтернативних напрямів розвитку, які визначаються як можливі сценарії розвитку та реалізації таких бізнес-процесів.

Сценарії послідовності дій визначають:

- конкретний процес, дію, регламент чи документ, що ініціює цей сценарій;
- межі виконання дій у рамках отримання конкретного результату цих дій;
- повноваження виконавця чи виконавців цих дій;
- доступні інформаційні джерела, які необхідні для виконання дій сценарію;
- документи, керівні впливи (доручення, команди), що створюють окремі дії сценарію;
- зв'язки з іншими сценаріями у рамках загального бізнес-процесу.

Сценарії послідовності дій формуються досить просто на основі побудови діаграм бізнес-процесів у нотації IDEF0. Використання такого підходу особливо зручно і часто використовується у разі опису складних багаторівневих процесів, що мають ієрархічну структуру.

При цьому, сценарії дій можуть бути відображені на різних рівнях ієрархії бізнес-процесу, що дає можливість надавати більш повну картину виконання дій як окремими виконавцями на нижніх рівнях ієрархії бізнес-процесу, так і групами виконавців у рамках структури загального бізнес-процесу верхнього рівня.

#### Питання до розділу 3

1. Як можна визначити поняття «сценарію»?
2. Як формуються рольові сценарії?
3. Які особливості рольових сценаріїв?
4. Які характерні особливості формування сценаріїв послідовності дій?
5. Які є особливості формування сценаріїв послідовності дій?
6. Які нотації використовуються для відображення сценаріїв?

## РОЗДІЛ 4. МЕТОДОЛОГІЇ ОПИСУ БІЗНЕС-ПРОЦЕСІВ

### 4.1. Поняття методу моделювання бізнес-процесу

Поняття методу моделювання базується на ряді наступних визначень, які описують основні поняття та взаємозв'язки між ними. Сукупність цих визначень дає можливість опису та визначення основних особливостей бізнес-процесів та сценаріїв с точки зору їх моделювання [1, 4, 5, 6, 7, 8, 10,].

Формування моделі бізнес-процесу – це складна задача, що потребує набору відповідних методів та засобів.

Для кожного об'єкту моделювання та мети побудови моделі обирають конкретну методику представлення процесів у вигляді стандартних блок-схем, діаграм, виконаних визначеним способом.

Метод створення схеми бізнес-процесу – найважливіша частина методології опису бізнес-процесів проекту організації.

Існує достатній перелік різних методів опису бізнес-процесів та сценаріїв, вибір яких визначається поставленими задачами та метою створення моделей бізнес-процесів.

Метод — це сукупність практичних і теоретичних прийомів, що дозволяють отримати рішення поставленої задачі, як спосіб досягнення мети.

Вибір відповідного методу опису залежить від цілей, які визначені аналітиком, що створює модель організації.

Структура методу опису бізнес-процесів (рис 4.1) складається таких основних елементів як:

- визначення;
- дисципліна;
- використання.

Визначення описує концепцію методу, мотивацію його використання, можливості та особливості та теоретичні засади на яких побудовано цей метод.

Дисципліна визначає його синтаксис та процедури, що необхідно виконувати у рамках цього методу.

Використання визначає специфіку використання методу та можливості його використання як окремого методу чи разом з якимось іншими методами, що доповнюють чи розширюють можливості використання цього методу.

## 4.2. Основні визначення методу моделювання бізнес-процесів

Бізнес-процес (процес) – керована, цілеспрямована послідовність взаємопов'язаних дій, що повторюються, і за визначеною процедурою, з урахуванням існуючих обмежень на використання ресурсів та стану оточуючого середовища, перетворюють вхідний продукт у вихідний з додаванням цінності для користувача/споживача цього бізнес-процесу.

Дія – це частина бізнес-процесу, виконання якої дає певний результат.

Декомпозиція – послідовна деталізація бізнес-процесу вищого рівня на бізнес-процеси нижчого рівня та/або операцій, виконання яких цілком забезпечує реалізацію вибраного БП.



Рис. 4.1 Структура методу опису бізнес-процесів

Модель бізнес-процесу – графічне, табличне, текстове чи символічне зображення бізнес-процесу або їх взаємозалежна сукупність, придатна для аналізу. Як правило, модель БП подається у форматі однієї зі стандартних мов опису.

Операція (робота) – елементарна складова частина бізнес-процесу, що виконується одним учасником за неподільний час з виділеними ресурсами та результатом якої є один матеріальний чи інформаційний вихід.

Підпроцес – бізнес-процес, який є складовою декомпозиції вищого рівня бізнес-процесу, що розглядається.

Сценарій (Сценарний процес) – частина бізнес-процесу, що визначає дії однієї ролі, чи показує послідовність дій для виконання однієї конкретної задачі.

Кожний бізнес-процес складається на нижньому рівні опису з одного чи кількох послідовних сценаріїв.

Роль – підрозділ або конкретний виконавець певної дії бізнес-процесу чи сценарію.

Функція – сукупність однорідних операцій (у тому числі таких, що входять до різних бізнес-процесів), які виконуються структурним підрозділом, посадовою особою чи роллю.

При моделюванні бізнес-процесів використовують два терміни «моделювання» та «опис», які на практиці не розрізняють, незважаючи на різне змістове навантаження.

Моделювання як і опис має на увазі створення схеми (діаграми) побудованої за певними вимогами з метою передачі інформації про діяльність об'єкту та її текстовий та табличний опис.

На практиці розглядають створення схем (діаграм) процесів за допомогою обраного методу.

Основою методів побудови опису бізнес-процесів є системний підхід, що ґрунтується на поняттях представлення об'єктів (чи елементів об'єктів) та зв'язків між ними.

Системний підхід будується на розгляді кожного конкретного бізнес-процесу як елементу системи, який визначає свої функції на основі взаємодії з іншими бізнес-процесами.

Як правило, об'єкт інформатизації розглядається як множина елементів (процесів, дій), що мають характерні властивості, та зв'язки між ними.

### **4.3. Методи опису бізнес-процесів**

Обов'язковим засобом первісного опису бізнес-процесів є текстовий опис у довільній формі. Як правило при текстовому описі дотримуються певних прийнятих правил.

Текстовий опис бізнес-процесів підприємства виконується за наступними характеристиками:

- повне найменування процесу;

- мета процесу (необхідний або бажаний результат);
- власник процесу (особа, що відповідає за перспективне планування, ресурсне забезпечення та ефективність процесу);
- керівник процесу (особа, відповідальна за поточне планування і здійснення процесу);
- нормативи процесу (документація, що містить показники норм, відповідно до яких здійснюється процес);
- входи процесу (матеріальні і інформаційні потоки, що поступають зовні і потребують перетворення);
- виходи процесу (результати перетворень);
- ресурси.

Наступним кроком є формальний опис, при якому вводяться спеціальні внутрішні форми, за допомогою процес описується через систему документів: таблиць документообігу, формулярів дій тощо.

Для досягнення бажаної ясності і точності в описі процесу використовують графічний опис.

Серед графічних засобів, які використовуються для структуризації бізнес-процесів, найбільш розповсюдженими є:

- дерево цілей бізнес-процесів;
- алгоритмізація бізнес-процесів;
- організаційні структура виконавців бізнес-процесів;
- графічне накладення бізнес-процесів на існуючу функціональну організаційну структуру підприємства;
- матриця відповідальності;
- структурний аналіз процесів (визначення основних клієнтів і постачальників, входи і виходи процесу).

Модель системного представлення організації (рис. 4.2), для якої проводиться побудова системи бізнес-процесів на початкових етапах може розглядатися, як сукупність:

- дерева функцій;
- потоків робіт;
- описів діяльності організації.

Під описами діяльності організації розуміються різноманітні документи, що визначають напрямки діяльності організації та формалізують методи та засоби виконання функцій, відповідних до діяльності, це та такі документи як:

- статуту;
- регламенти;
- стандартів;
- законодавчі акти;
- посадові інструкції;
- установчі та інших документи.

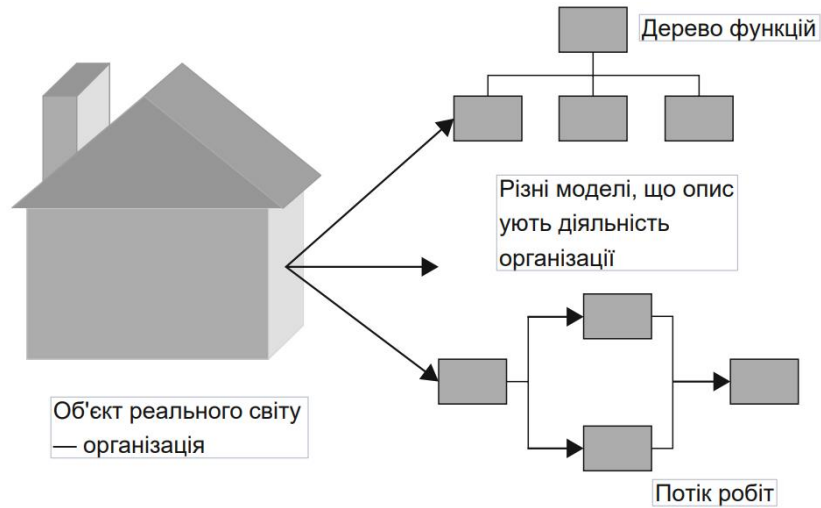


Рис. 4.2 Модель системного представлення організації

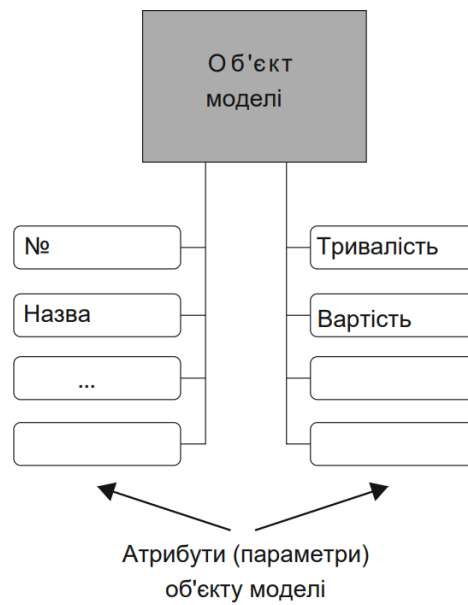


Рис. 4.3 Модель 2 системного представлення організації

Крім того, для повноти опису необхідно визначити основні атрибути моделі системного представлення організації (рис 4.3), які визначають усі основні впливові на бізнес-процес параметри об'єкту, для якого будується система бізнес-процесів.

Важливим елементом при цьому є визначення певних меж значень атрибутів (скопів) та їх відносний вплив одного на інші.

Загальні вимоги до методології моделювання визначаються тим, що методологія реалізується через конкретні технології та підтримуючі їх стандарти, методики та інструментальні засоби, які забезпечують відображення бізнес-процесів та сценаріїв всього життєвого циклу об'єкту моделювання.

Технологія моделювання визначається як сукупність трьох складових:

- покрокових описів дій, яка визначає послідовність операцій моделювання бізнес-процесів та сценаріїв;
- критерії і правила, використовувані для оцінки результатів побудови моделей бізнес-процесів та сценаріїв дій;
- нотації (графічні і текстові засоби), що використовуються для опису системи бізнес-процесів та їх декомпозицій.

Цілі формування моделей процесів організації мають бути чіткими та зрозумілими, тому необхідно розробити конкретні вимоги до цих моделей та порядок їхнього практичного використання.

Особливу увагу слід звернути на обмеження ступеня деталізації та глибини опису бізнес-процесів, кількість об'єктів, зв'язків та їх атрибутів. Подробиці моделі мають бути адекватними поставленим цілям та розв'язуваним завданням.

Вибір методології повинен ґрунтуватися на чіткому розумінні їх можливостей і недоліків, а також цілей використання створюваних моделей бізнес-процесів.

#### **4.4. Моделі структурні, функціональні, інформаційні**

Залежно від властивостей об'єктів з точки зору теорії моделювання моделі БП поділяються на:

- структурні моделі;
- функціональні моделі;
- інформаційні моделі.

Структурна модель (structural model) описує об'єкт моделювання з точки зору його складу та взаємозв'язку частин (елементів системи) між собою і з зовнішнім середовищем.

Зв'язки між елементами можуть бути: фізичні; логічні; інформаційні.

Сукупність елементів системи та зв'язків між ними утворюють структуру системи.

Структурні моделі найчастіше існують у формі різноманітних структурних схем.

Функціональні моделі (functional model) описують процеси, що відбуваються в об'єкті моделювання.

Інформаційна модель (information model) – система даних про об'єкт та опис потоків даних в процесі його функціонування.

#### **Питання до розділу 4**

1. Що таке методологія моделювання бізнес-процесів?
2. Які складові входять до методології моделювання бізнес-процесів?
3. Які використовуються методи опису бізнес-процесів?
4. Які загальні вимоги до методології моделювання бізнес-процесів?
5. Які можна визначити методи опису бізнес-процесів?
6. Які види моделей бізнес-процесів ви знаєте?

## РОЗДІЛ 5. НОТАЦІЇ ОПИСУ БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ

### 5.1. Основні методології опису процесів

Серед існуючих методологій опису бізнес-процесів найбільш розповсюджені три групи (таблиця. 5.1) [1, 3, 4, 8, 10]:

- методології опису моделей структурного типу;
- методології опису потоків робіт та процесів;
- методології опису потоків даних.

Таблиця 5.1 Моделі бізнес процесів

Моделі бізнес-процесів	
Структурні моделі (управління)	Нотація IDEF0, ARIS, Casewise, MS Visio
Моделі потоків робіт	Нотація IDEF3, ARIS, Casewise, Business Studio, MS Visio
Моделі процесів, що виконуються	BPMN, UML
Моделі потоків даних	DFD, UML

### 5.2. Методології сімейства ICAM – IDEF

IDEF (I-CAM DEFinition або Integrated DEFinition) – методологія сімейства ICAM (Integrated Computer-Aided Manufacturing) для вирішення завдань моделювання складних систем об'єктів, що дозволяють відображати та аналізувати моделі діяльності широкого спектру складних систем у різних розрізах. При цьому широта та глибина обстеження процесів у системі визначається самим розробником, що дозволяє не перевантажувати створювану модель зайвими даними.

Методології IDEF (SADT – Structured Analysis and Design Technique) було розроблено у межах запропонованої ВПС США програми комп'ютеризації

промисловості – ICAM, як можливість ефективного обміну інформацією між усіма фахівцями – учасниками програми ICAM.

З понять методології IDEF пов'язано виникнення основних ідей BPR (Business Process Reengineering).

Методології IDEF досить широко поширені і зараз у США та у Європі, бо дають можливість досить повно та на різних рівнях формалізації описувати бізнес-процеси.

Найчастіше використовуються наступні нотації методології сімейства IDEF:

IDEF0 – Function Modeling (Моделювання бізнес-процесу) — методологія функціонального моделювання і графічного опису процесів, призначена для формалізації і опису бізнес-процесів (рис. 5.1, 5.2). IDEF0 орієнтовано на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання (Workflow). Дає можливість показати зв'язки між окремими сценаріями та їх ієрархічну підпорядкованість.

IDEF3 – Process Description Capture – методологія документування процесів, що проходять в системі (наприклад, в організації), описує сценарії дій та послідовність операцій для процесів нижніх рівнів.

IDEF5 – Ontology Description Capture – методологія опису онтології при дослідженні складних систем та структур даних. Використовується в побудові сценаріїв для моделювання пошуку та обробки інформації в інформаційно-аналітичних системах.

DFD – Data Flow Diagram – це нотація, призначена для моделювання систем з точки зору зберігання, обробки та передачі інформації. Використовується для моделювання та аналізу сценаріїв обробки даних.

Види функцій IDEF0:

- діяльність;
- процес;
- операція;
- дія.

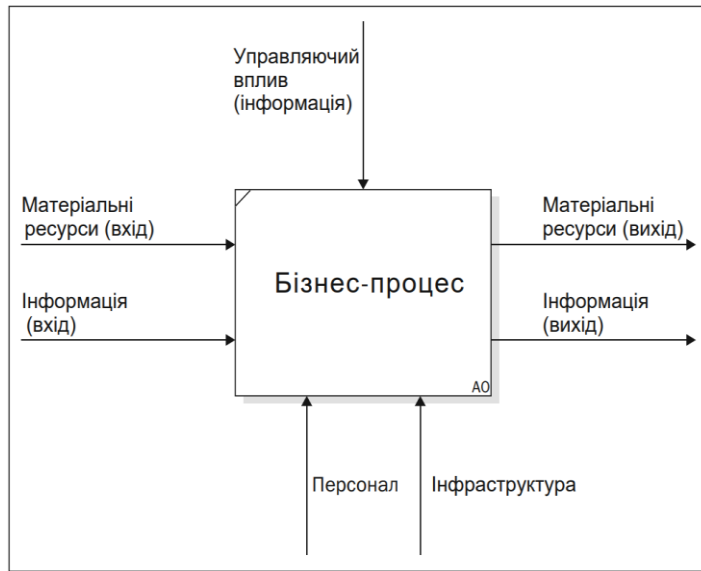


Рис. 5.1 Загальна модель бізнес-процесу

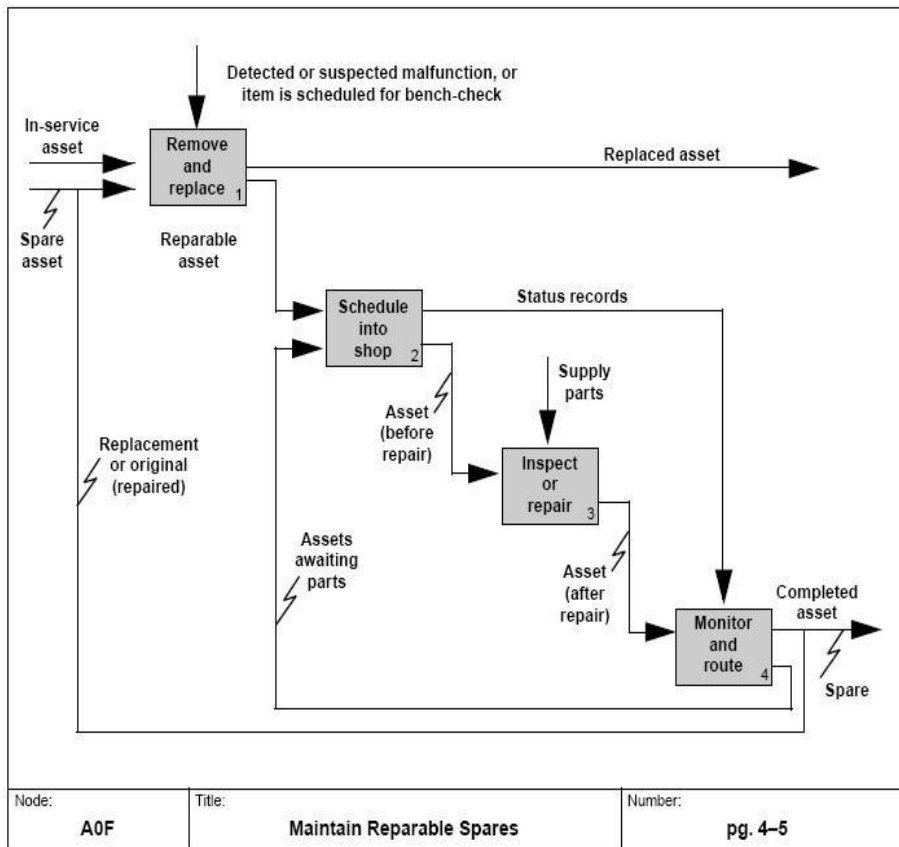


Рис. 5.2 Зразок діаграми IDEF0

IDEF0 – найбільш популярна нотація побудови бізнес-процесів заснована на методології структурного аналізу SADT, та призначена в першу чергу для опису процесів верхнього рівня.

Методологія IDEF0 опису бізнес-процесів полягає в описі дій за допомогою діаграм чотирьох типів:

1. Контекстна діаграма – її прийнято вважати головною діаграмою, оскільки націлена на зображення основної функції та її взаємодію із зовнішнім середовищем.
2. Діаграмами декомпозиції вважаються другорядними або дочірніми. Описують складові основної функції.
3. Діаграма дерева вузлів висловлює залежність функцій між собою.
4. Діаграми для експозиції – розроблені для зображення окремих частин системи, створена для вираження оптимальної точки зору бізнес-процесу.

### 5.3. Переваги та недоліки методології IDEF0

Аргументи переваги та недоліків методології IDEF0 розглянуто в таблиці 5.2.

Таблиця 5.2 Переваги та недоліки IDEF0

Переваги	Недоліки
<ol style="list-style-type: none"> <li>1. Повнота опису бізнес-процесу (управління, інформаційні та матеріальні потоки, зворотні зв'язки).</li> <li>2. Комплексність при декомпозиції (мігрування та тунелювання стрілок).</li> <li>3. Можливість агрегування та деталізації потоків даних та інформації (розподіл та злиття стрілок).</li> <li>4. Наявність жорстких вимог, що забезпечують отримання моделей процесів стандартного виду.</li> <li>5. Простота документування процесів.</li> <li>6. Відповідність підходу до опису процесів в IDEF0 стандартам ISO 9000:2005</li> </ol>	<ol style="list-style-type: none"> <li>1. Складність сприйняття (суттєва кількість стрілок).</li> <li>2. Велика кількість рівнів декомпозиції.</li> <li>3. Складність узгодження декількох процесів представлених в різних моделях однієї і тієї організації.</li> </ol>

## 5.4. Методологія IDEF3

Нотація IDEF3 - найважливіша після IDEF0 і призначена для опису потоків робіт (Workflow Modelling) (рис. 5.3). Протягом тривалого часу IDEF3 широко використовувалася для створення моделей бізнес-процесів організації на нижньому рівні – при описі робіт, що виконуються в підрозділах та на робочих місцях.

Цю нотацію було взято за основу при створенні методики опису процесів ARIS (Architecture of Integrated Information Systems) — «розширеного ланцюжка процесу, керованого подіями».

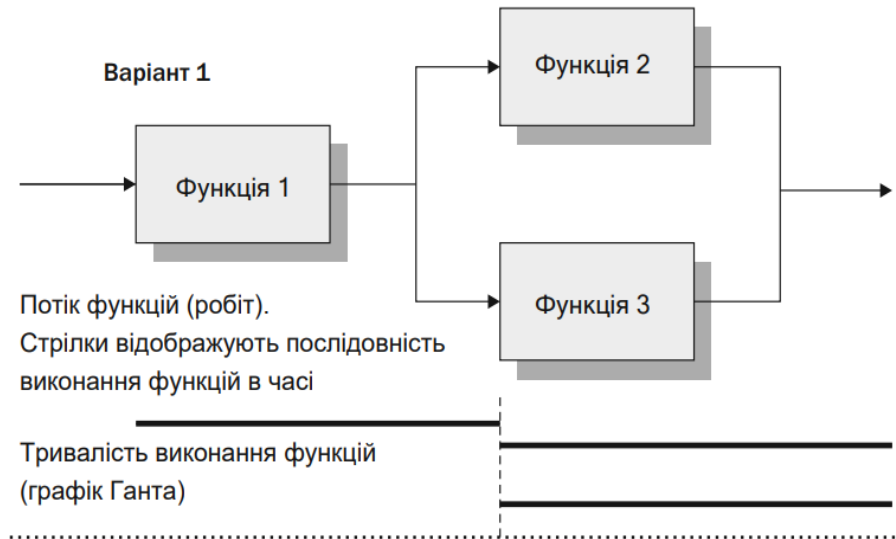


Рис. 5.3 Схеми представлення робіт блоками

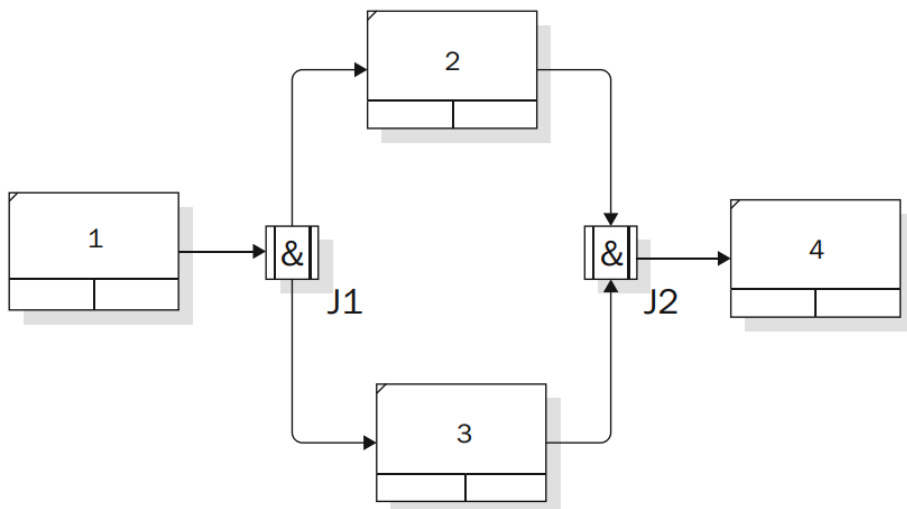


Рис. 5.4 Правила побудови стрілок

Для подолання неоднозначності представлення опису бізнес-процесів в нотації IDEF3 додано додаткові об'єкти:

- логічне «І»;
- логічне «або»;
- виключне логічне «або».

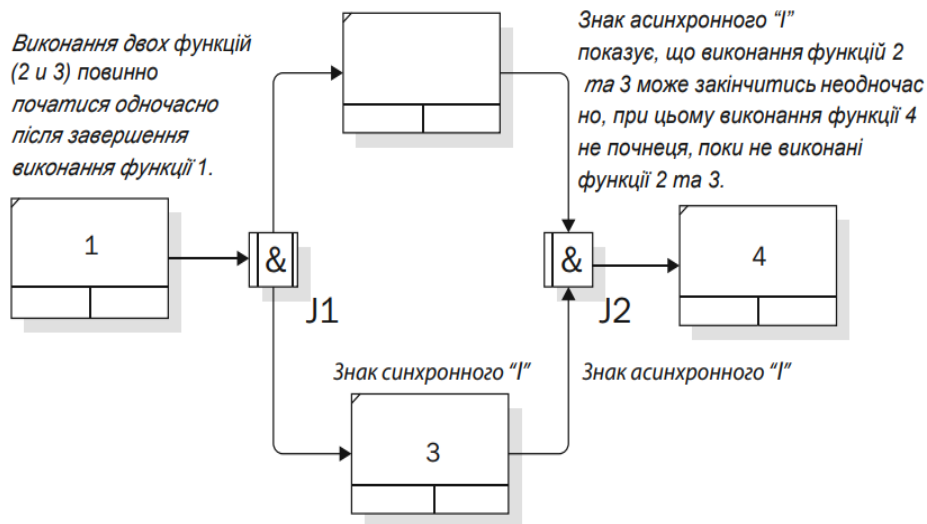


Рис. 5.5 Правила побудови логічних операторів

На відміну від нотації IDEF0, у нотації IDEF3 сторони чотирикутника, що зображує функцію (роботу, процес), не використовуються для прив'язки входів різного типу. Більше того, у чотирикутник входить і виходить лише одна стрілка. В іншому випадку правила побудови діаграм у IDEF3 будуть порушені.

У нотації IDEF3 логічні оператори (рис. 5.4, 5.5) можуть бути синхронними та асинхронними.

При декомпозиції процесів IDEF3 не відбувається мігрування і тунелювання стрілок. Аналітик має сам дбати про зв'язність моделювання процесу, коректність декомпозиції.

## 5.5. Висновки з IDEF3

Нотацію IDEF3 доцільно застосовувати у разі відносно простих процесів на нижньому рівні декомпозиції, тобто лише на рівні робочих місць.

У цьому випадку схема процесу може служити основою для створення документів, які регламентують роботу виконавців. Процес у нотації IDEF3 досить простий.

За допомогою цієї нотації досить складно створювати комбіновані моделі, в яких би поєднувалися описи потоків робіт та процеси управління ними.

У нотації IDEF3 досить просто та зрозуміло можна описувати сценарії послідовності робіт, як групи робіт бізнес-процесів нижнього рівня моделей.

## 5.6. Моделювання процесів в нотації DFD

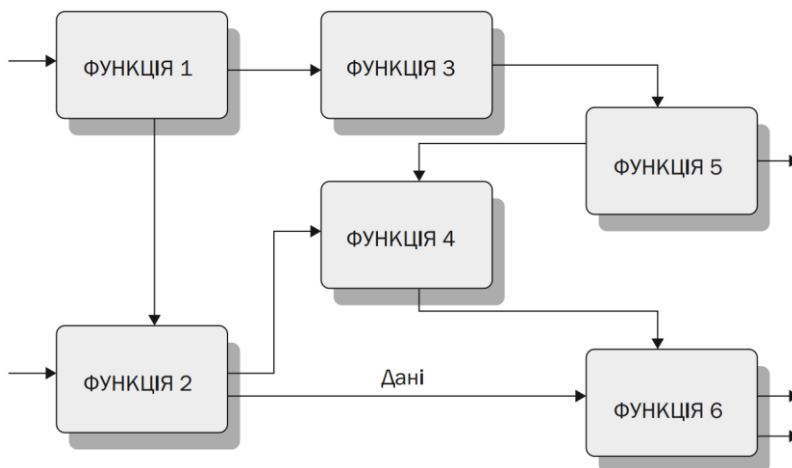


Рис. 5.6 DFD (Data Flow Diagram)

DFD (Data Flow Diagram) (рис. 5.6) - діаграми потоків даних (інформації) опису процесів.

Діаграми цього типу містять, як правило, два типи графічних об'єктів: чотирикутники та стрілки. Перші описують функції (роботи, процеси), другі - потоки даних між ними.

На діаграмі DFD функції зазвичай розташовуються зліва направо у порядку, що відповідає послідовності їх виконання у часі, подібно нотації IDEF3, хоч це не обов'язково. До опису процесів в DFD застосовують типові правила декомпозиції. В нотації DFD сторони чотирикутника не мають того ж значення, що у IDEF0.

## 5.7. DFD

У кожному великому підрозділі виконується ряд різних бізнес-процесів. Часто у цих процесів існують різні внутрішні та зовнішні клієнти. Саме тому DFD схеми описують тільки потоки даних, що передаються між функціональними підрозділами, але нічого не говорять про реально виконувані бізнес-процеси як на рівні підрозділів, так і на рівні об'єкту інформатизації у цілому.

DFD потрібна для опису реально існуючих в організації потоків даних. Описи можуть створюватися як за процесною, так і за функціональною структурою. У першому випадку ми отримуємо моделі бізнес-процесів у форматі DFD, у другому - схему обміну даними між підрозділами.

Нотація DFD може ефективно застосовуватися для опису сценаріїв збору та обробки інформації на підприємствах та організаціях. (Збір інформації про результати здачі сесії та інше).

## 5.8. Практичне призначення DFD

Створені моделі потоків даних організації можуть бути використані при вирішенні таких завдань, як:

- визначення існуючих сховищ даних (документів, файлів, СУБД);
- визначення даних, необхідних для виконання кожної функції процесу;
- підготовка до створення моделі структури даних організації.

Широко використовується для опису та моделювання при розробці та реінжинірингу систем документообігу та контролю виконання доручень (рис. 5.7) для опису та сценаріїв збору та обробки інформації.

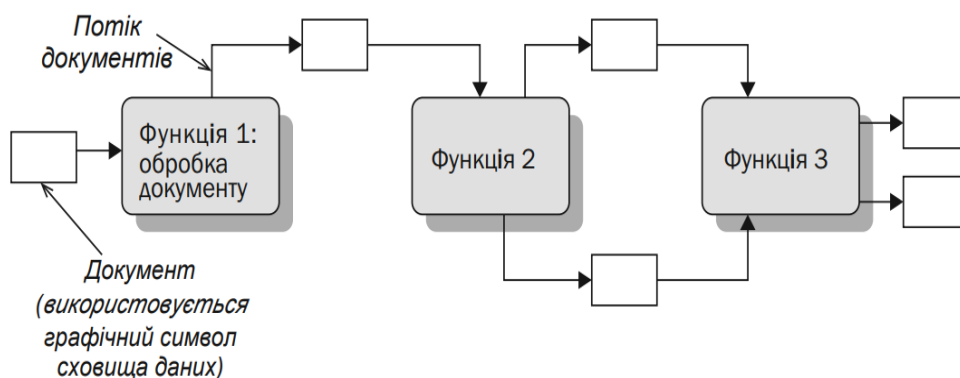


Рис. 5.7 Потік документів в DFD

## 5.9. Види моделей в методології ARIS

Методологія ARIS (рис. 5.8) включає кілька різних нотацій для опису діяльності організації з різних точок зору. У методологію інтегровано існуючі стандарти та специфікації опису процесів та даних, наприклад IDEF3, ERD, DFD, UML і т. ін.

До найбільш значущих і практично використовуваних нотацій ARIS відносяться:

- нотація Value-added Chain Diagram (діаграма ланцюжка доданої вартості);
- нотації eEPC (Extended Event-driven Process Chain) – розширений ланцюг процесів, керований подіями;
- PCD (Process Chain Diagram) (діаграма ланцюжка процесу);
- нотація Organizational Chart (організаційна діаграма);
- нотація Function Tree (дерево функцій);
- нотація Product Tree (дерево продуктів).

## 5.10. Концепція ARIS щодо опису організації

Підхід методології ARIS до опису процесів ґрунтується на розгляді діяльності організації з чотирьох точок зору:

1. погляд на організаційну структуру;
2. погляд на дані (потоки та структуру);
3. погляд на функції (функціональні ієрархії);
4. погляд на контроль та управління (зведені моделі бізнес-процесів).

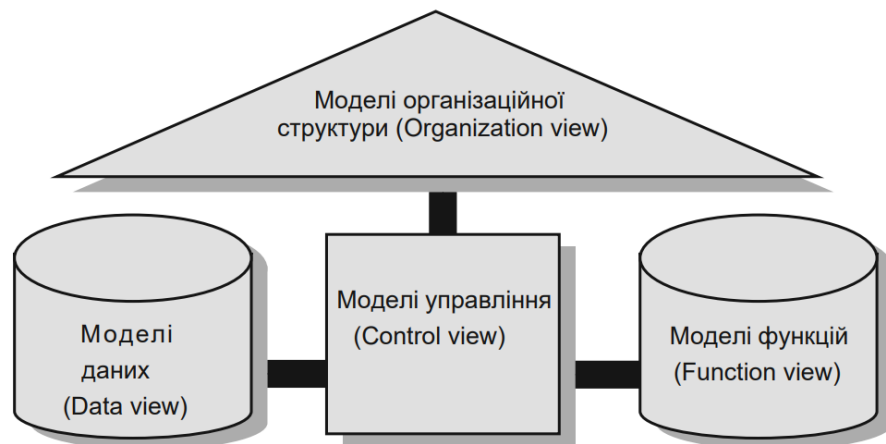


Рис. 5.8 Концепція моделі ARIS

## 5.11. Переваги ARIS

Привабливість методології ARIS базується на її комплексності, яка проявляється у взаємозв'язку моделей, побудованих у різних нотаціях.

Методологія ARIS дозволяє описувати діяльність об'єктів інформатизації з різних точок зору, при цьому отримані моделі до певної міри пов'язані між собою.

Основні недоліки комплексного підходу ARIS:

- для реалізації, інструментальне середовище ARIS, досить дороге та складне у використанні, хоча існує і безкоштовна, спрощена версія цього продукту під назвою ARIS Express, але вона в основному демонстраційна;
- важко реалізовувати на практиці, так, як передбачає великий об'єм ресурсів (людських, матеріальних та фінансових) протягом тривалого часу.

## **5.12. BPM та BPMS**

BPM (Business Process Management, управління бізнес-процесами) – це концепція процесного управління організацією, що поєднує в собі ідеологію та програмне забезпечення для управління бізнес-процесами.

Концепція BPM розглядає діяльність об'єктів інформатизації як набір бізнес-процесів і чітко відповідає на всі питання про кожен з них (наприклад, де і коли виконується процес, хто відповідальний і інше).

BPM – це концепція процесного управління організацією, яка розглядає бізнес-процеси як особливі ресурси підприємства, що безперервно адаптуються до постійних змін.

Основні принципи даної концепції – це зрозумілість і прозорість бізнес-процесів. Досягається це за рахунок їх моделювання з використанням формальних нотацій, використання програмного забезпечення для симуляції, моніторингу, моделювання та аналізу бізнес-процесів, динамічного перестроювання моделей бізнес-процесів силами персоналу і засобами програмних систем.

BPMS (Business Process Management System або Business Process Management Software, система управління бізнес-процесами) – це програмне забезпечення для втілення BPM-концепції у життя, її технологічна частина.

## **5.13. Нотація BPMN**

BPMN (Business Process Model and Notation) (рис. 5.9) - це мова моделювання бізнес-процесів. За допомогою моделювання засобами BPMN ми можемо описати будь-які за складністю бізнес-процеси, для подальшої інформатизації.

Також BPMN – це система умовних позначень (нотація) і модель для опису та підготовки до автоматизації бізнес-процесів.

Нотація BPMN з'явилася у 2004 році. Вона орієнтована на опис так званих виконуваних процесів, які підтримуються системами автоматизації операційних процесів — BPM (Business Process Management).

BPMN виділяє два основних компоненти:

1. BPM (моделювання бізнес-процесів) — це середовище, в якому безпосередньо виконується моделювання.

2. BPMS (система моделювання бізнес-процесів) – це інструменти для виконання створених моделей.

Це може бути Bizagi, Comundo, ELMA та інші. Але це досить дорогі та складні системи.

Bizagi Process Modeler — безкоштовний та зручний застосунок, що використовується для опису діаграм і документування процесів на основі нотації BPMN.

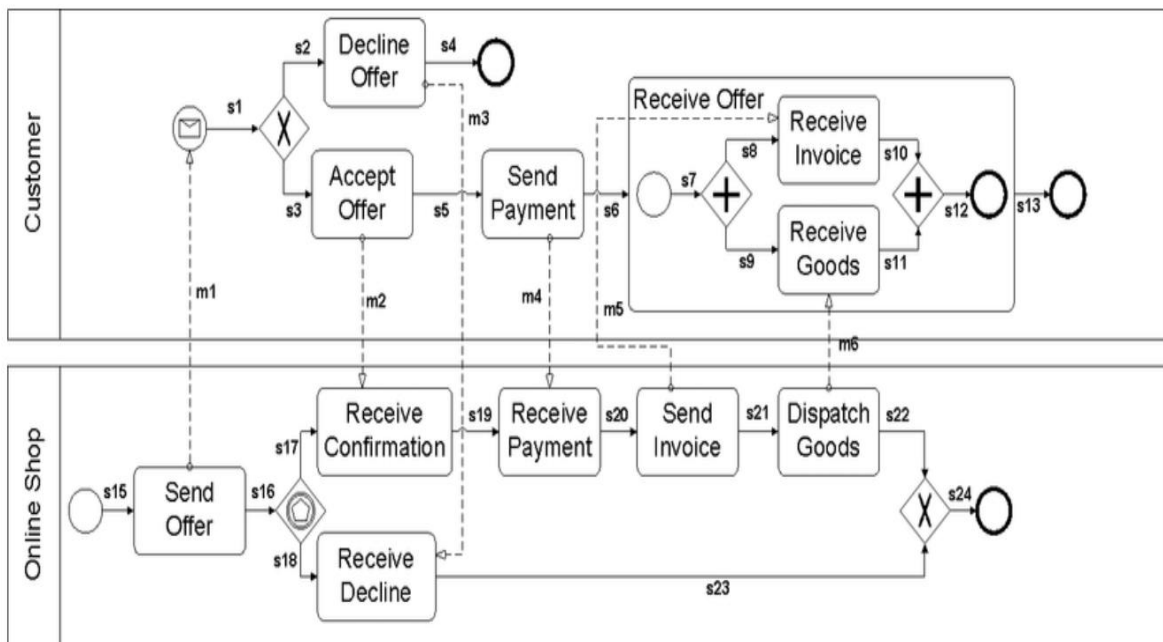


Рис. 5.9 Діаграма BPMN

IDEF3 + DFD + Swim Lane BPMN

## 5.14. Елементи нотації BPMN

В BPMN (рис. 5.10) - мова опису бізнес-процесів базується на таких базових об'єктах:

1. Event – Подія (подія, яка сталася в описі процесу).
2. Activity – Дії (задачі, що потрібно виконати).
3. Gateway - Шлюзи або Розвилки (керуючий вузол при розгалуженні процесів).
4. Flow – Потік (в BPMN ключовими для побудови моделі процесу є елементи потоку).

5. Data – Дані (елемент, що вказує, які дані та документи необхідні для початку дії або які результати завершені дії).
6. Artefact – Артефакти (об'єкти, які не є діями і не мають прямого відношення до дій).
7. Swimlane - "плавальні доріжки« (розташовані або по горизонталі, або по вертикалі та використовуються для угруповання процесів чи завдань).
8. Pool (Пул) – набір (це об'єкт, що описує один процес на діаграмі).
9. Association - асоціація – призначена для з'єднання інформації та артефактів з іншими елементами діаграми.

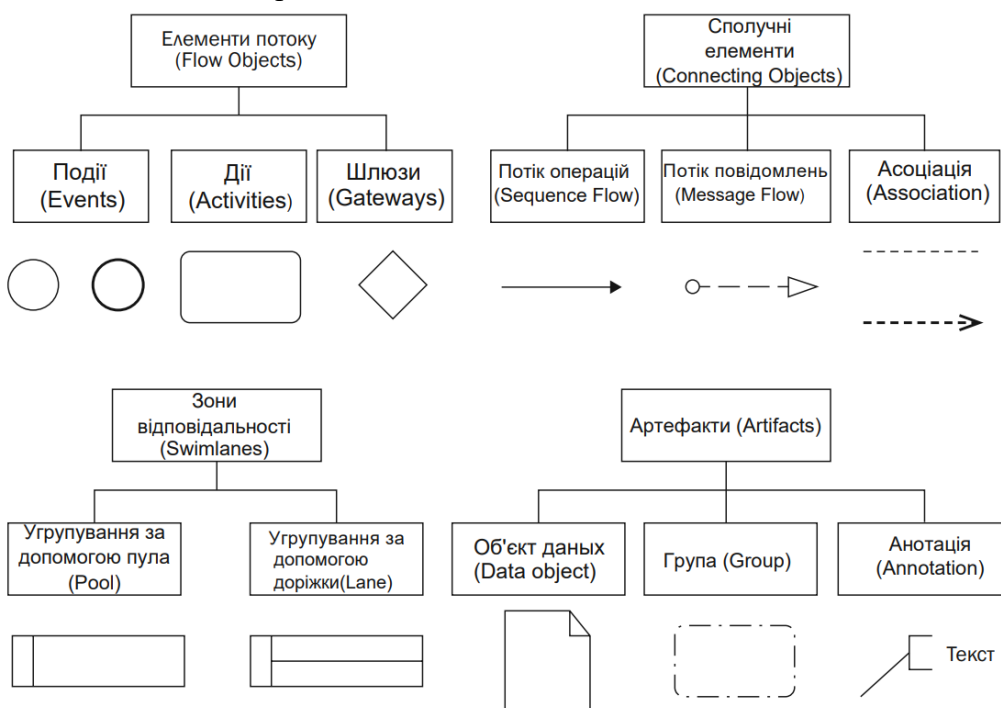


Рис. 5.10 Об'єкти нотації BPMN

## Питання до розділу 5

1. Які є основні типи методологій опису процесів?
2. У чому суть методології сімейства ICAM-IDEF?
3. Яке призначення нотацій IDEF0-IDEF5?
4. Яка особливість методології IDEF0?
5. Які переваги та недоліки методології IDEF0?
6. У чому призначення методології IDEF3?
7. Які можливості моделювання процесів в методології DFD?

8. Яка концепція методології ARIS?
9. В чому особливість концепції BPM?
10. Які основні елементи нотації BPMN?

## РОЗДІЛ 6. ОПИС БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ З ВИКОРИСТАННЯМ БЛОК-СХЕМ ПРОЦЕСІВ.


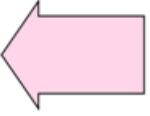
### 6.1. Опис процесів з використанням блок-схем

Найпростіший, але практично важливий спосіб опису бізнес-процесів, – методика складання блок-схем. Цей підхід має багато спільного з графічними мовами опису алгоритмів програмного забезпечення (таблиця 6.1). З погляду методології, формування блок-схем проводиться як і у нотації IDEF3, хоча для спрощення символи логіки можуть опускатися [1, 3].

Опис процесів за допомогою блок-схем має одну істотну перевагу – це простота і доступність сприйняття керівниками та спеціалістами підприємства. Витрати на навчання виконавців читання блок-схем мінімальні. Крім того, для їх формування не потрібні спеціалізовані, дорогі програмні продукти.

Форма подання блок-схем підходить для документування процесів і створення регламентуючої документації: сценаріїв виконання робіт, описів процесів, посадових та робочих інструкцій.






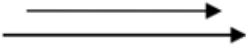

Таблиця 6.1 Об'єкти блок схем

№	Тип об'єкта/ з'єднувача	Опис/коментарі	Символ об'єкта/ з'єднувача
1	2	3	4
1	Вхід – інформаційні, матеріальні або фінансові потоки	Об'єкт використовується для зображення входу БП. Якщо входом БП є вихід іншого БП, то найменування цього об'єкта повинно бути унікальним і співпадати (збігатися) з найменуванням виходу іншого БП. Зображується зеленим кольором.	
2	Вихід - інформаційні, матеріальні або фінансові потоки	Об'єкт використовується для зображення виходу БП. Якщо вихід БП є входом іншого БП, то найменування цього об'єкта повинно бути унікальним і співпадати (збігатися) з найменуванням входу іншого БП. Зображується рожевим кольором.	

Продовження табл. 6.1

1	2	3	4
3	Дія	Об'єкт відповідає БП або операції. Ім'я кожного блоку має бути унікальним та однаковим для всіх діаграм, на яких цей об'єкт присутній. Зображується оранжевим кольором  Наприклад: «Візування документа»	
4	Дія	Об'єкт відповідає БП або операції. У верхній клітинці вказується найменування БП / операції. В нижній - підрозділ товариства або виконавець. Зображується оранжевим кольором  Наприклад: Верхня клітинка - «Візування документа»;  Нижня клітинка - «Відділ кадрів»	
5	Дія (декомпозиція дій)	Об'єкт відповідає БП або операції, що має декомпозицію в цій моделі. Зображується оранжевим кольором	
6	Зовнішня дія	Об'єкт відповідає БП або операції, що є зовнішньою відносно процесу/операції, що описується.  Зображується червоним кольором	
7	Вибір «Виключаюче АБО»	Правило розгалуження процесу. В об'єкті вказується умова вибору варіанта. На вихідних стрілках вказуються параметри, за якими обирається той чи інший варіант. Зображується бузковим кольором Наприклад: «Рахунок сплачений: так/ні»	

Продовження табл. 6.1

1	2	3	4
8	Вибір. Тригер	<p>Правило розгалуження процесу. В об'єкті вказується умова вибору варіантів. На вихідних стрілках вказуються параметри, за якими обирається той чи інший варіант. Зображується бузковим кольором</p> <p>Наприклад: «Тип скарги: загальна/технічна/фінансова».</p> <p>«Тип клієнта: корпоративний/масовий/оператор»</p>	
9	Інформаційний об'єкт. Автоматизована система	<p>Об'єкт відповідає автоматизованій системі. На блоці вказується реальна назва АС. на стрілочці - інформація, яка надходить з/до АС. Зображується сірим кольором Наприклад: База «Парус»</p>	
10	Інформаційний об'єкт. Документ	<p>Об'єкт відповідає офіційному документу на будь-якому носії (наказ, лист, бланк заявки тощо) або будь-якій інформації, що може виходити за межі цього БП.</p> <p>Для найменування необхідно використовувати їх реальне найменування та тип документа. Зображується блакитним кольором</p> <p>Наприклад: «Запит щодо...», «Наказ щодо створення ГУП»</p>	
11	Посилання	<p>Об'єкт використовується для відображення переходу між блоками (в межах однієї сторінки або між сторінками схеми) для однієї діаграми</p>	
12	Коментар	<p>Об'єкт використовується для відображення коментарів до блоків діаграми</p>	 <p><u>Примітки</u></p>
13	З'єднувачі	<p>Напрямок передачі дій. Назву об'єкта на стрілці слід вказувати тільки у разі, коли вона не очевидна</p>	
14	З'єднувач з інформаційними ресурсами та БД	<p>Напрямок передачі інформації документів та БД. Назву об'єкта на стрілці слід вказувати тільки у разі, коли вона не очевидна</p>	

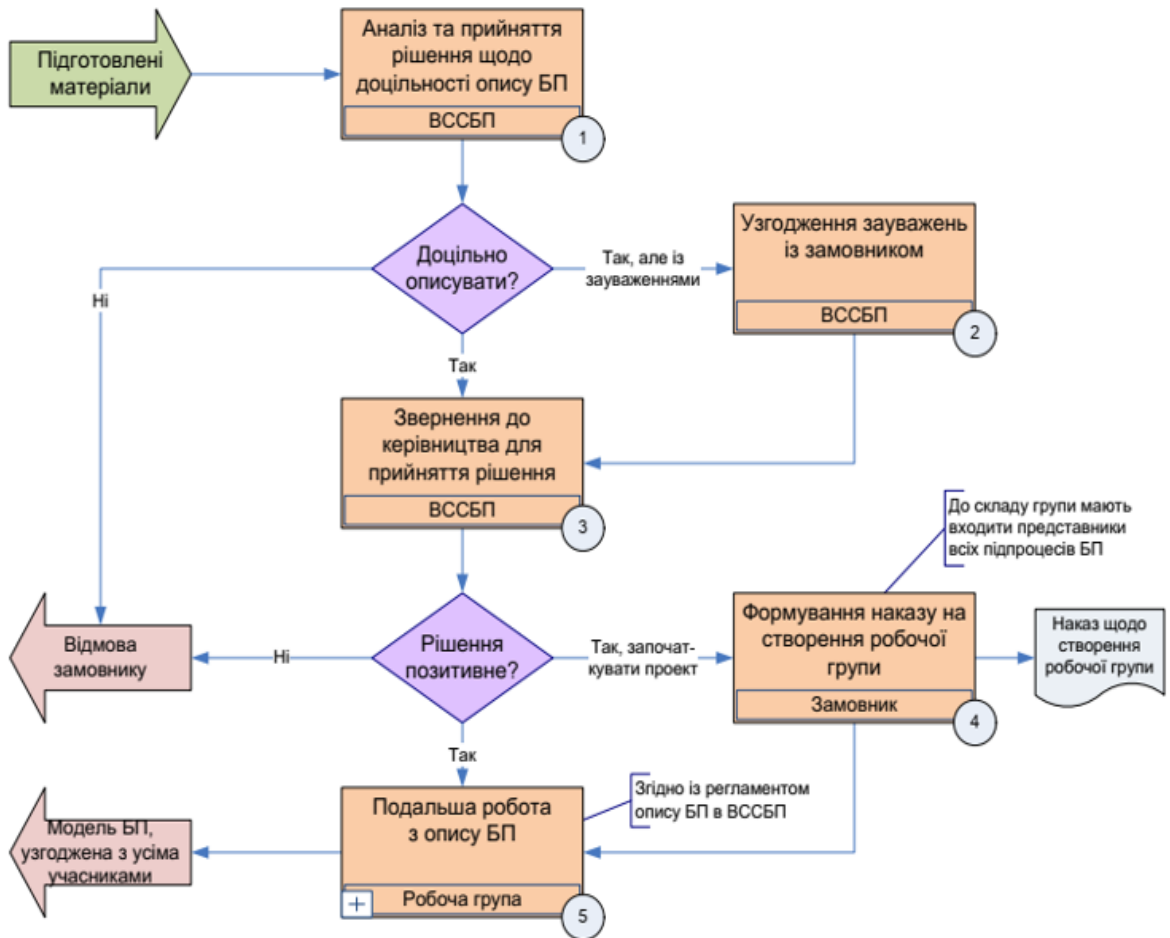


Рис. 6.1 Приклад бізнес процесу опису бізнес-процесу

## 6.2. Swim Lane Diagram (SLD)

Swim Lane Diagram (SLD) (рис. 6.3) – різновид графічної нотації, що використовується для кращого розуміння БП з точки зору відповідальності виконавців за конкретний БП/операцію та відображення сценаріїв дій кожної з ролей (виконавців учасників БП), а також відображення сценаріїв поведінки окремих ролей для виконання завдань у рамках моделі БП.

Правила формування схем БП у форматі Swim Lane Diagram (SLD):

1. SLD являє собою графічне зображення бізнес-процесу у вигляді декількох горизонтальних або вертикальних доріжок.
2. Кожна з доріжок відображає підпроцеси/операції, які виконуються одним конкретним виконавцем (учасником БП) чи групою, що не поділяється.

3. На останній доріжці зображуються інформаційні системи (бази знань/даних, архіви, документи), які використовуються для виконання операцій у БП.
4. Правила формування та умовні позначення в SLD такі самі, як і в графічній нотації, крім застосування об'єктів, які визначають виконавця.
5. У символі об'єкта не зазначається виконавець.

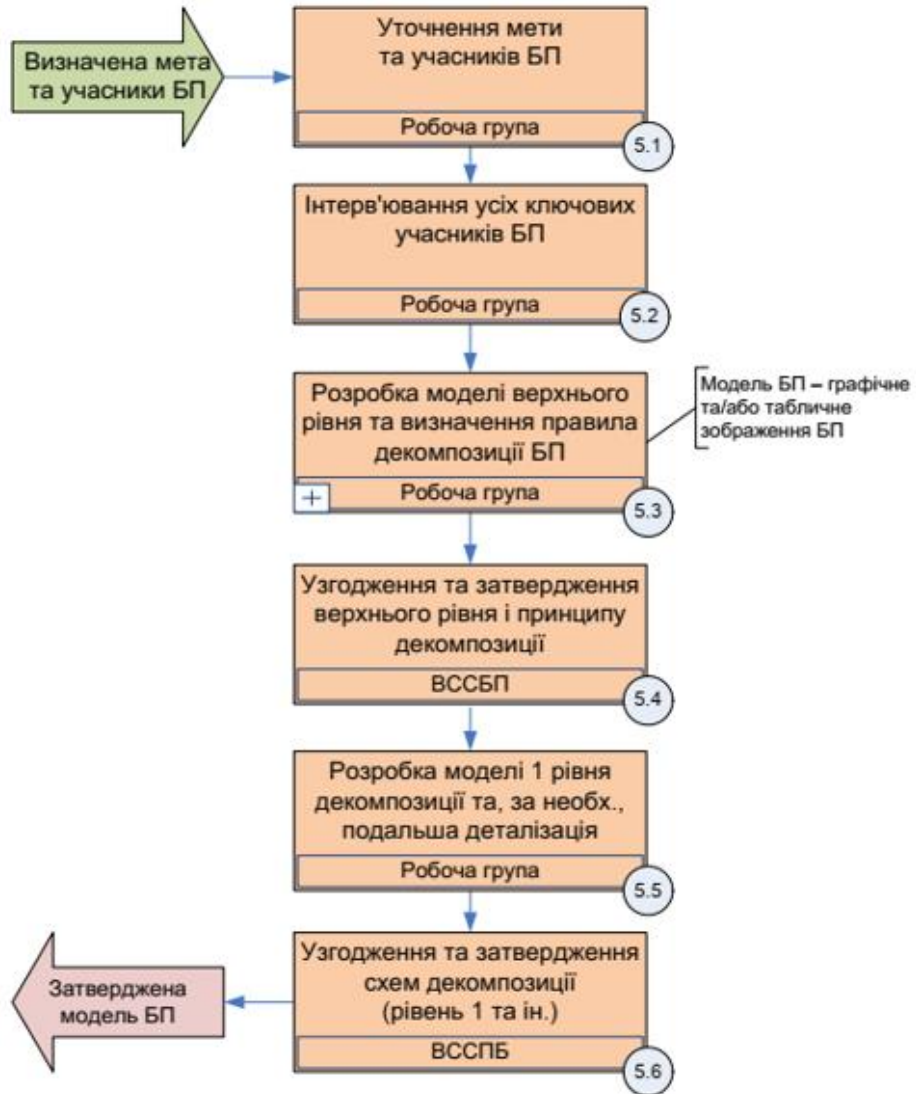


Рис. 6.2 Декомпозиція блоку № 5 бізнес-процесу опису бізнес-процесу

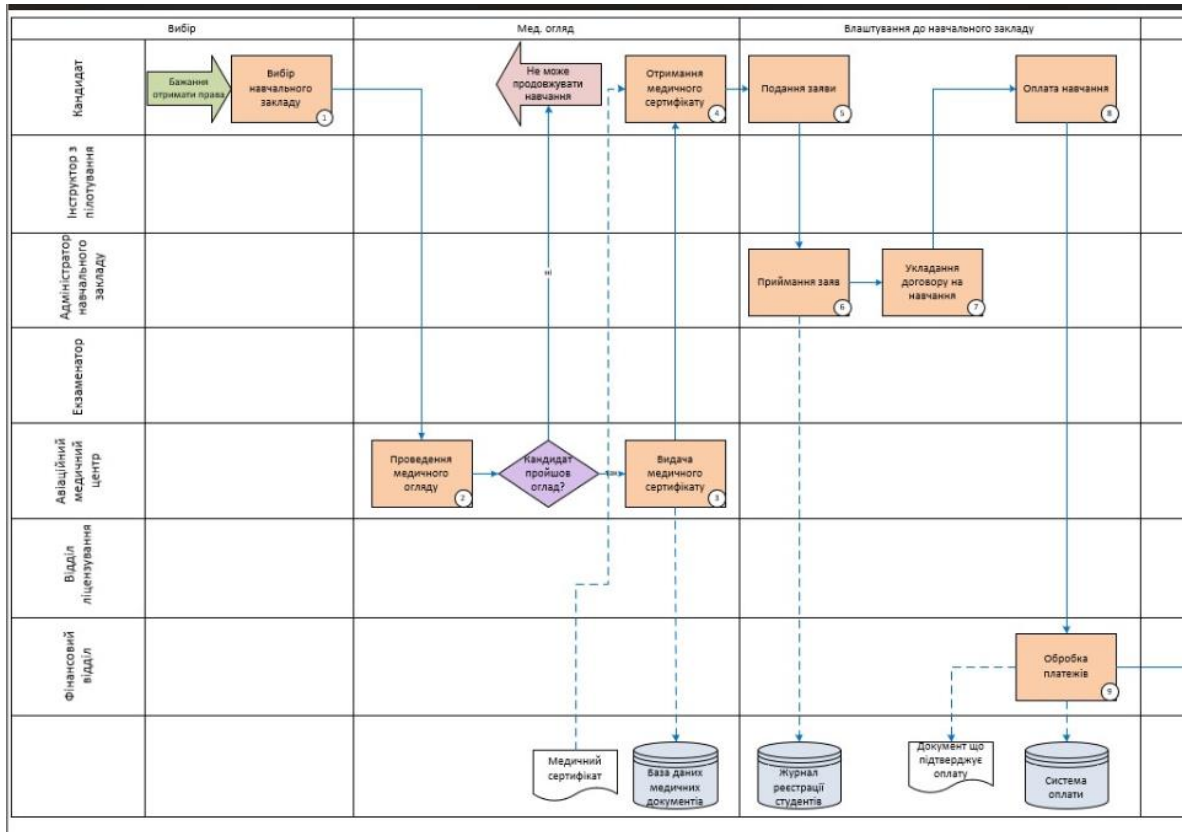


Рис. 6.3 Приклад побудови бізнес-процесу у форматі SLD

Як правило, для повноти опису треба визначити функції ролей учасників бізнес-процесу та їх дії (таблиця 6.2, 6.4). Приклади можливих дій за ролями приведено у табл. 6.3.

Таблиця 6.2 Ролі та функції учасників і бізнес-процесів

№	Роль	Функції сценарію	№ блоку на схемі	Пов'язані інформативні ресурси (документи, таблиці та інше)	Джерела та форма надання інформації (електронна, паперова)
1		1			
		2			
		...			
...		1			
		2			
		...			

Таблиця 6.4 Приклад дій за ролями

Назва	Ролі приклади (>4)	Дії приклади (> 15)
Отримання прав на управління спортивним літаком	Здобувач, менеджер курсів, бухгалтер, інструктор, екзаменатори	Пошук, оплата, навчання, льотна практика, здача теоретичних та практичних екзаменів
Покупка квартири у кредит	Фізична особа, працівник агентства, менеджер байку, представник компанії продавця	Пошук, перемовини з агентством та банком, отримання довідок, рішення банк, оформлення покупки
Придбання земельної ділянки у кредит	Покупець, працівник агентства, працівник банку, працівник Держкомзему, продавець	Пошук, перемовини, оцінка, оформлення договорів, перевірка прав продавця на ділянку, отримання кредиту, оплата
Отримання сертифікату з іноземної мови	Отримувач, працівник курсів, викладач, бухгалтер, сертифікаційна комісія, директор курсів	Пошук, перемовини, тестування, оплата, навчання, внутрішні заліки, сертифікація по рівнях знань
Надання автомобіля у оренду	Власник автомобіля, працівник агентства, нотаріус, оцінювач, орендатор	Перемовини з агентством, оцінка умов, оформлення договору, оплата
Придбання туристичної путівки у кредит	Покупець, працівник агентства, банк, страхова компанія	Вибір агентств, перемовини, вибір туру, перемовини з банком, оплата, оформлення страховки
Закупівля обладнання та послуг по встановленню локальної мережі	Покупець, юрист, представники компанії продавця, працівники компанії виконавця робіт	Пошук, перемовини з продавцем та виконавцем, оцінка, оформлення договорів, оплата
Здача офісу в оренду через сайт агентства по нерухомості	Орендодавець, працівник агентства, директор, бухгалтер, орендатор	Вибір агентств, визначення умов розміщення на сайті, оцінка стану офісу, складання договору, оплата
Закупівля товарів для магазину	Менеджер по закупівлям, товарознавець, працівник торгової бази, продавець, бухгалтери	Вибір товару та бази, перевірка наявності продукції, оцінка вартості та якості продукції, підготовка договору та оплата
Купівля квитка на концерт онлайн	Покупець, працівник агентства, банк, касир, кур'єр	Пошук, перемовини з агентством та театром, оцінка варіантів, оплата, доставка

1. Визначити основні дії за сценаріями (таблиця 6.4).

Таблиця 6.4 Основні дії за ролями та сценаріями

№	Роль	Дії за сценарієм	Вхідні параметри	Вихідні параметри
1		1.Блок №...		
		2. Блок №...		
		...		
2		1.Блок №...		
		2. Блок №...		
		...		
...		1.Блок №...		
		2. Блок №...		
		...		

### Питання до розділу 6

1. Чим характерний опис бізнес-процесів за допомогою блок-схем?
2. У чому переваги опису бізнес-процесів за допомогою блок-схем?
3. Як формується ієрархія бізнес-процесів з застосуванням блок-схем?
4. Які основні об'єкти опису процесів з використанням блок-схем?
5. Які основні можливості та особливості Swim Line Diagram?
6. Які є правила формування схем БП у форматі Swim Lane Diagram?

## РОЗДІЛ 7. МЕТОДОЛОГІЯ SADT. НОТАЦІЯ IDEF0

### 7.1. Підхід IDEF0

Методологія IDEF0 заснована на підході, розробленому Дугласом Т. Россом на початку 70-х років і отримав назву SADT (Structured Analysis & Design Technique - метод структурного аналізу та проектування).

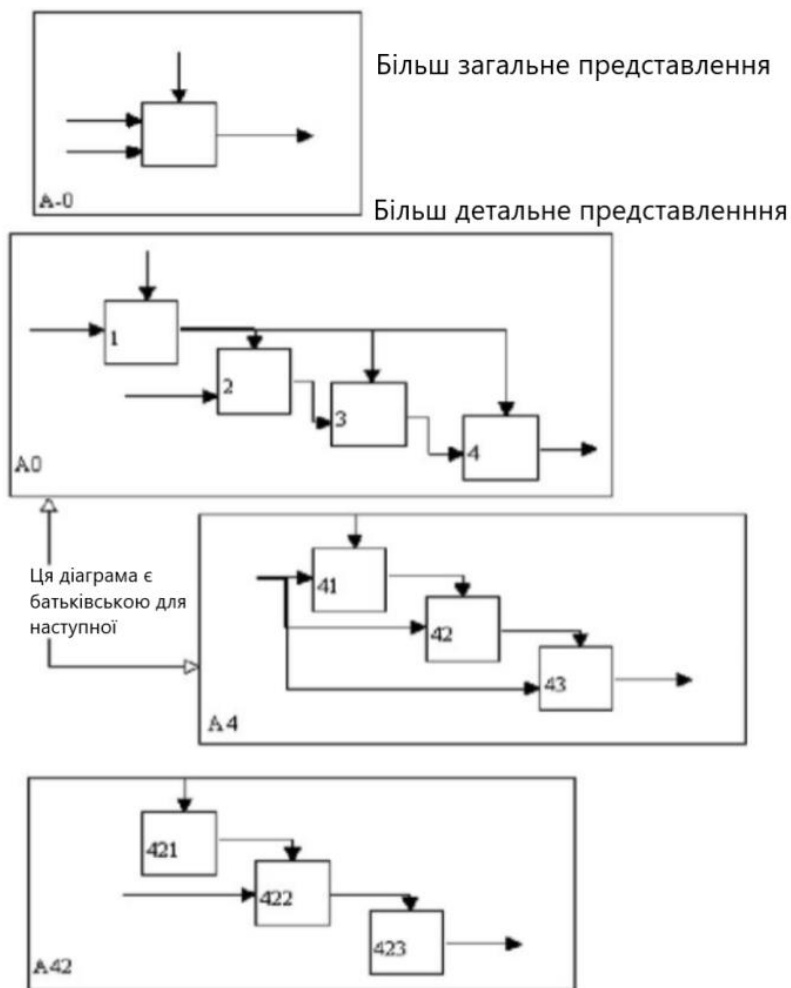


Рис. 7.1 Метод структурного аналізу та проектування

У своєму початковому вигляді IDEF0 включав як визначення мови графічного моделювання (синтаксис і семантика), так і опис комплексної методології розробки моделей [1, 8 - 10].

IDEF0 – методологія функціонального моделювання. За допомогою наочного графічного мови IDEF0, система функцій об'єктів інформатизації постає перед

розробниками і аналітиками у вигляді набору взаємопов'язаних функцій (функціональних блоків - в термінах IDEF0).

Особливістю методології IDEF0 є визначення функціональної моделі об'єкту інформатизації у вигляді ієрархічної моделі (рис. 7.1). Це значно спрощує розуміння складу окремих компонентів моделі об'єктів інформатизації.

Методологія IDEF0 може бути використана для моделювання широкого спектру автоматизованих та не автоматизованих систем.

Для нових інформаційних систем IDEF0 може використовуватися:

- для визначення вимог та характеристик функцій;
- для розробки деталізації функцій;
- для контролю відповідності компонентів ІС, що розробляються, функціям, що визначені.

Для існуючих систем IDEF0 може використовуватися:

- для аналізу функцій (as is);
- для їх декомпозиції;
- для реінжинірингу функцій (to be);
- для подальшої інформатизації функцій у рамках програмно-технічних систем, що розробляються.

## **7.2. IDEF0-моделі**

IDEF0-моделі (рис. 7.2) складаються з графічних діаграм та тексту. Ці документи мають перехресні посилання один на одного.

Графічна діаграма – головний компонент IDEF0-моделі, що містить блоки, стрілки, з'єднання блоків та стрілок та пов'язані з ними відносини. Блоки представляють основні функції об'єкта, що моделюється.

Ці функції можуть бути розбиті (декомпозовані) на складові і представлені у вигляді більш складних діаграм; процес декомпозиції триває до того часу, поки об'єкт описаний лише на рівні деталізації, необхідному задля досягнення цілей конкретного проекту.

Діаграма верхнього рівня моделі (рис. 7.4) дає найбільш загальний або абстрактний опис предмета, представленого моделлю. За цією діаграмою слідує ряд дочірніх діаграм (рис. 7.5, 7.6), що дають більш детальну інформацію про об'єкт.

Кроки побудови IDEF0. Побудова моделі IDEF0 полягає у виконанні наступних дій:

1. збір інформації про об'єкт, визначення його меж;

2. визначення мети та точки зору моделі;
3. побудова та декомпозиція діаграм;
4. оцінка повноти опису, рецензування та коментування.

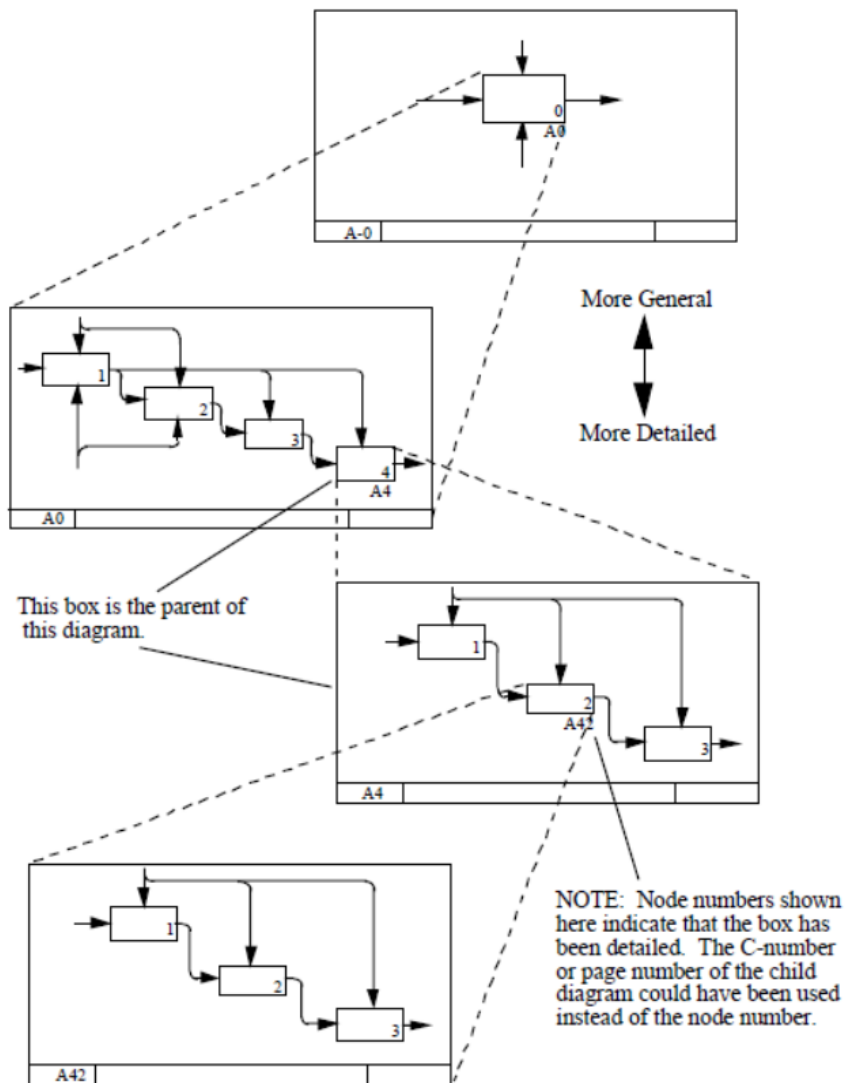


Рис. 7.2 Структура моделі IDEF0

### 7.3. Правила побудови діаграм IDEF0

Синтаксис діаграм визначається наступними правилами:

1. Діаграми складаються з блоків (рис. 7.7) та дуг.
2. Блоки відображають функції.
3. Кількість блоків обмежена – від 3 до 6-7.
4. Блоки мають домінування (домінуючий в лівому верхньому куті).

5. Дуги відображають набори об'єктів, що передаються між блоками.
6. Дуги зображують взаємозв'язок між блоками.

У IDEF0 розрізняють п'ять типів зв'язків робіт (рис. 7.3, таблиця 7.1):

1. Вихід – вхід.
2. Вихід – управління.
3. Вихід – механізм виконання.
4. Вихід – зворотній зв'язок на управління.
5. Вихід – зворотній зв'язок на вхід.



Рис. 7.3 П'ять типів зв'язків робіт у IDEF0

Таблиця 7.1 Тракткування зв'язків робіт у IDEF0

Традиційний переклад	Правильний переклад
Input — Вхідні	Input (Ввід)
Output — Вихідні	Output (Вихід)
Control — Управління	Control (Контроль)
Mechanism — Механізм	Mechanism (Механізм)
	Call (Виклик)*

\* *Call (Виклик)* – різновид стрілки *Mechanism (Механізм)*

Стрілка входу – стрілка (Arrow) може нести значення інформаційних, фінансових чи інших ресурсів, документів на паперовому та електронному носії, що ініціюють процес чи дію.

Характеризує вхідні дані чи параметри.

Завжди приєднується до функціонального блоку зліва.

Стрілка виходу – стрілка (Output) відображає відходи процесу, операції, діє, як звітність, продукцію або послугу, перетворені дані (в тому числі інформаційні).

Характеризує результати дії чи процесу.

Завжди виходить з процесу, операції, дії, функції (функціонального блоку) праворуч.

Стрілка управління – визначає клас стрілок (Control), які в IDEF0 відображають управління, тобто умови, при виконанні яких вихід блоку буде правильним. Стрілка відображає (контролює) вплив зовнішнього середовища на процес у вигляді міжнародних і вітчизняних стандартів, внутрішніх стандартів підприємства, посадових або робочих інструкцій, технічної документації, законодавчих актів і так далі.

Характеризує обмеження процесу.

Завжди приєднується до функціонального блоку зверху.

Стрілка механізму – визначає клас стрілок механізми (Mechanism):, які відображають засоби IDEF0, що використовуються для виконання функції; включає спеціальний випадок стрілки виклику. Символізують різні ресурси, такі як, співробітників, програмне забезпечення, обладнання, засоби зв'язку, тобто все те, що бере участь в процесі.

Характеризує ресурси необхідні для виконання дії чи процесу.

Завжди приєднуються до процесу знизу.

Стрілка виклику – це вид стрілки механізму (Call), який означає звернення з блоку даної моделі (або частини моделі) до блоку іншої моделі (або іншої частини тієї ж моделі) та забезпечує зв'язок між моделями або між різними частинами однієї моделі.

Характеризує зв'язки необхідні для виконання дії чи процесу.

Завжди приєднуються до процесу знизу.

## **7.4. Правила стрілок IDEF0**

Кожна сторона функціонального блоку повинна мати стандартне відношення блоку/стрілки:

- a) Стрілки, входів, повинні підходити до лівої сторони блоку.
- b) Стрілки контролю повинні підходити до верхньої сторони блоку.
- c) Стрілки, що виводять, повинні відходити від правої сторони блоку.
- d) Стрілки механізму (крім стрілок виклику) повинні вказувати вгору та підключатися до нижньої сторони блоку.
- e) Стрілки виклику механізму повинні вказувати вниз, підключатися до нижньої сторони блоку, і позначатися посиланням на блок, що викликається.

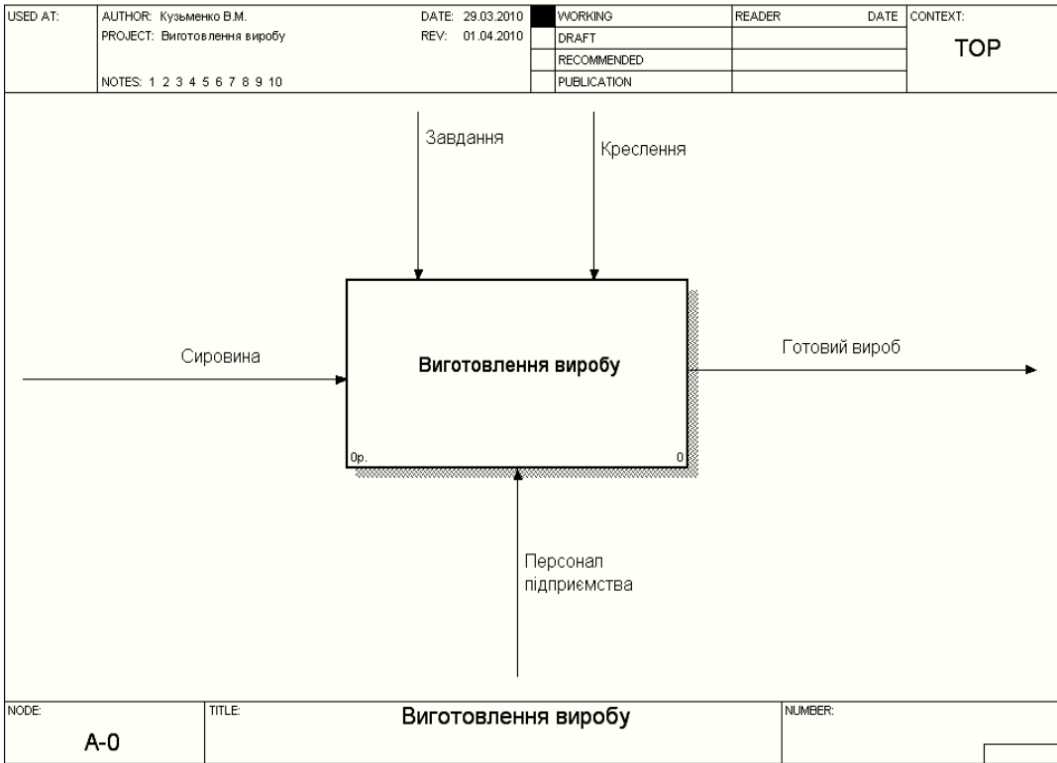


Рис. 7.4 Контекстна діаграма – діаграма найвищого рівня

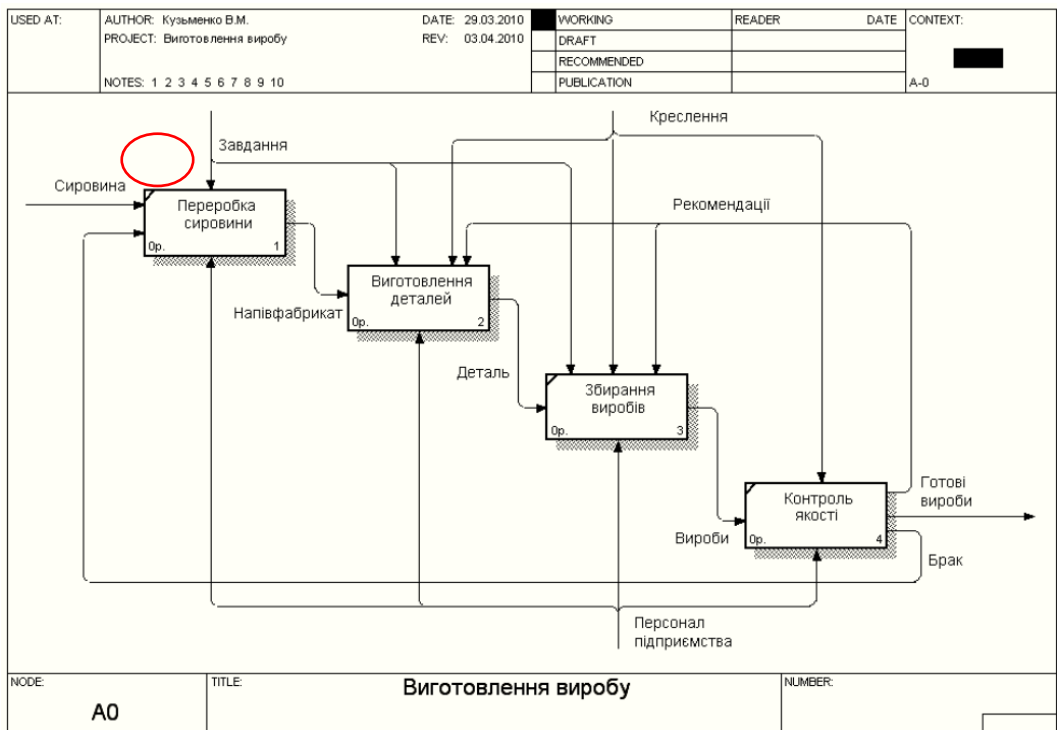


Рис. 7.5 Діаграма декомпозиції першого рівня

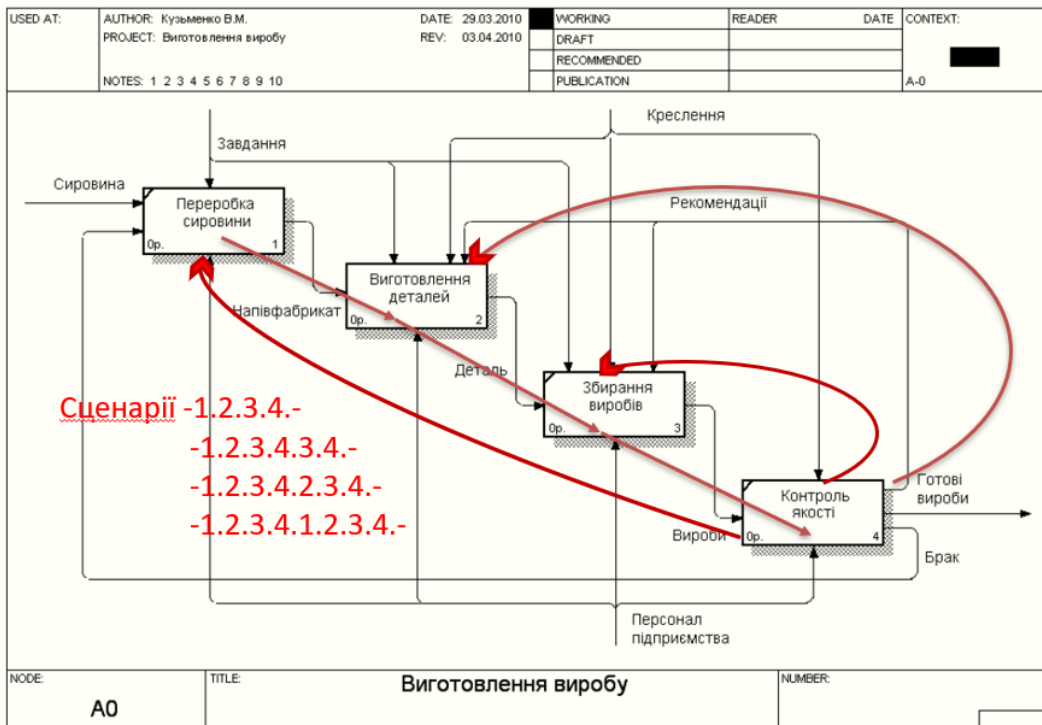


Рис. 7.6 Діаграма декомпозиції IDEF0 та можливі сценарії дій (Цикли у сценаріях – це результат зворотних зв'язків)

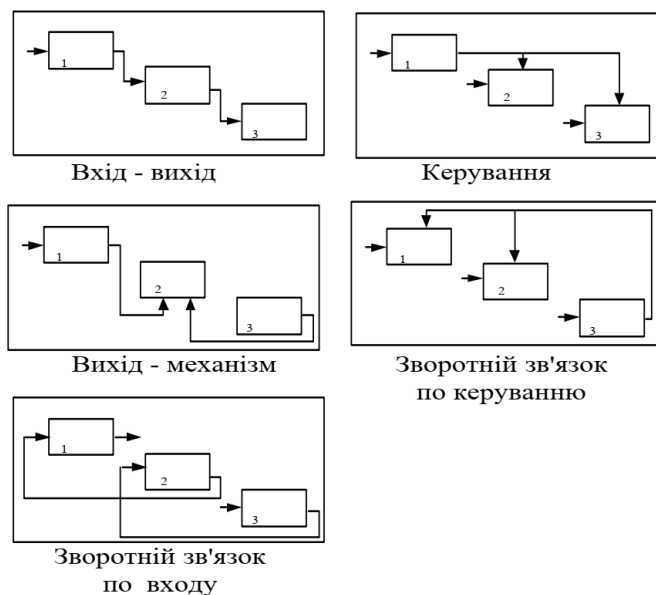


Рис. 7.7 Типи зв'язків блоків функціональної моделі IDEF0

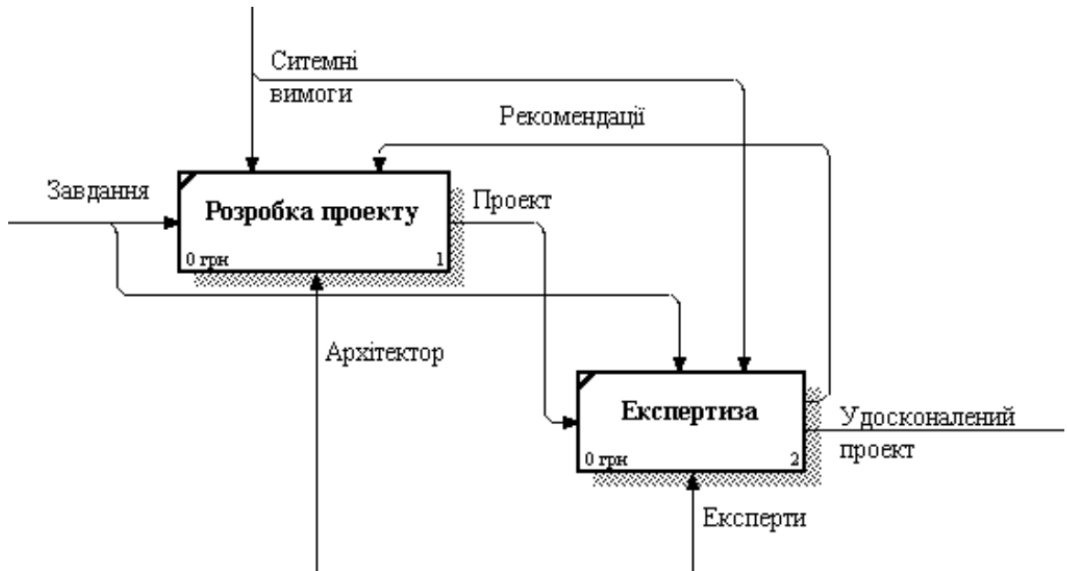


Рис. 7.8 Зворотній зв'язок по керуванню – циклічні процеси

Код ICOM (рис. 7.9): Аббревіатура (Input Введення, Control Управління, Output Висновок, Mechanism – Механізм), код, що забезпечує відповідність граничних стрілок дочірньої діаграми зі стрілками батьківського блоку; використовується для посилань.

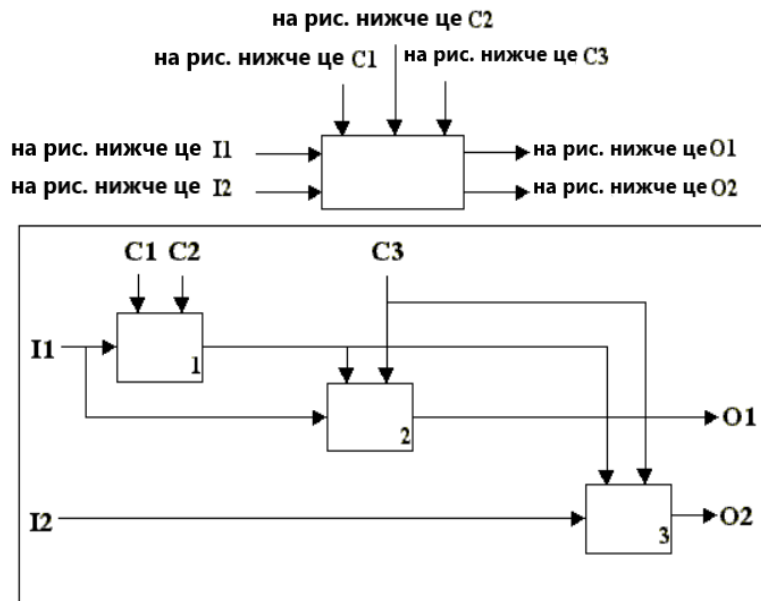


Рис. 7.9 Код ICOM

Коди ICOM гарантують стикування діаграм. Записують на діаграмі декомпозиції.

## 7.5. Розгалуження стрілок – паралельні процеси

Розгалуження стрілки (рис. 7.10, 7.11), що зображується у вигляді ліній, що розходяться, означає, що весь вміст стрілки або його частина може з'явитися в кожному відгалуженні стрілки. Розгалуженням стрілок є приєднання до будь-якої точки стрілки початку іншої стрілки.

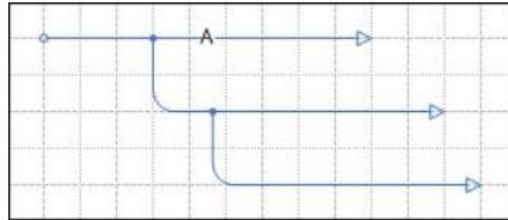


Рис. 7.10 Стрілки. Паралельні процеси

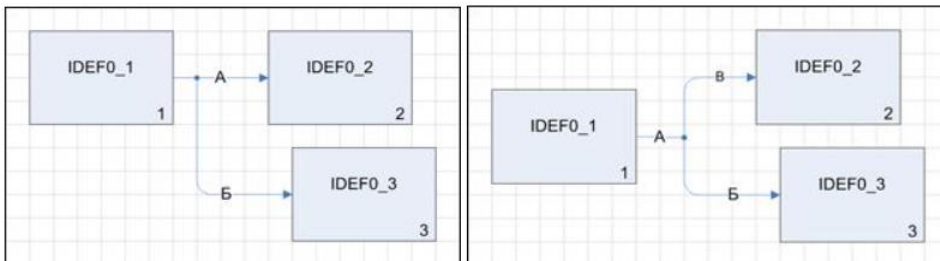


Рис. 7.11 Паралельні процеси

## 7.6. Злиття стрілок

Злиття стрілок (рис. 7.12), що зображується як лінії, що сходяться разом, вказує, що вміст кожної гілки потрапляє в результуючу стрілку. Злиттям стрілок є приєднання до будь-якої точки стрілки кінця іншої стрілки. Кількість стрілок, що приєднуються, не обмежена.

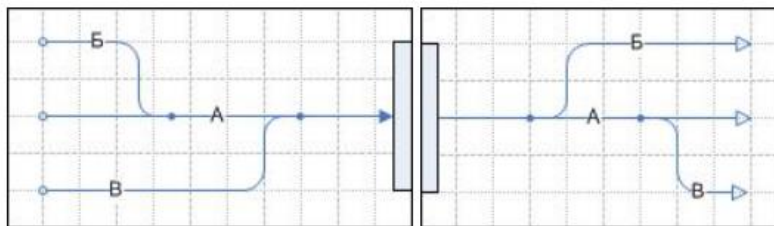


Рис. 7.12 Злиття стрілок

Черговість приєднання стрілок одна до одної значима: від неї залежить порядок іменування стрілок і успадкування об'єктів стрілки. Основною стрілкою називається стрілка, до якої приєднується стрілка-сегмент.

## 7.7. Міграція та тунелювання стрілок.

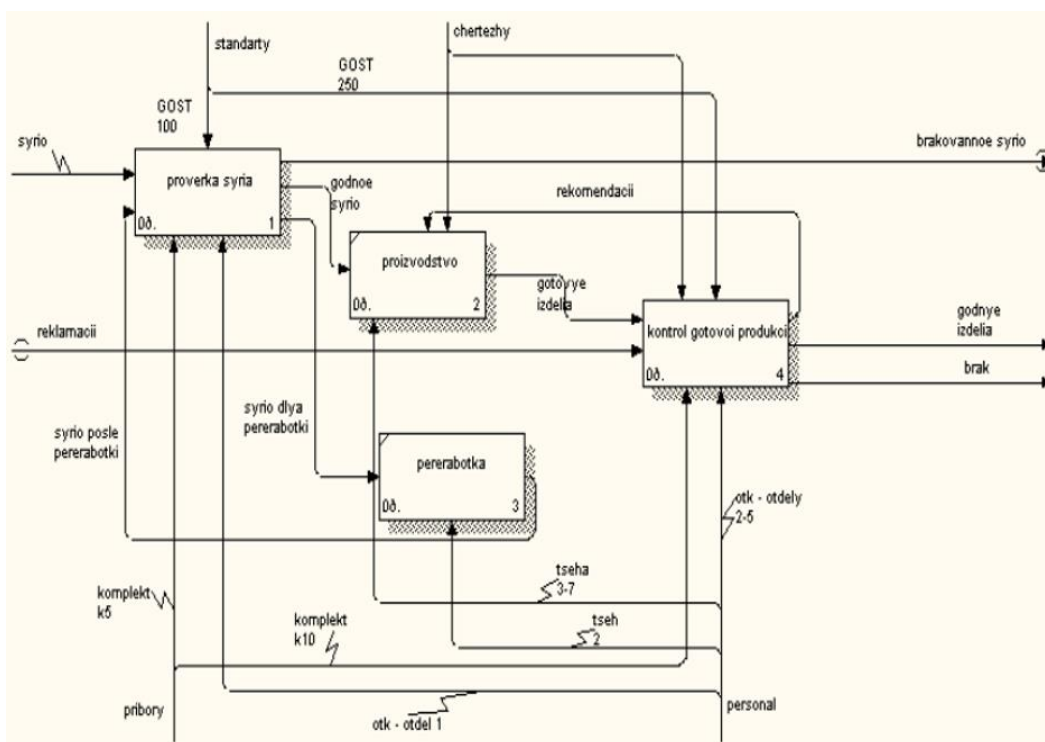


Рис. 7.13 Тунелювання стрілок

При першому кроці декомпозиції з'являється схема процесу, на якому показані стрілки, які не входять до жодного чотирикутника – це стрілки «мігрували» на рівень вниз. Їх необхідно з'єднати з конкретними функціями, використовуючи механізм розгалуження та злиття стрілок.

Необхідно відзначити, що всі стрілки, показані на верхньому рівні, будуть показані і на нижньому рівні, таким чином зберігається пов'язаність під час моделювання бізнес-процесів – підпроцеси виявляються пов'язаними з процесами верхнього рівня та навпаки.

Стрілка, укладена в квадратні чи круглі дужки (□ чи ○) (рис. 7.13), з'являється, коли ми показуємо на діаграмі нову, нижчого рівня, стрілку якої немає на діаграмі верхнього рівня. Для стрілок, що входять до діаграми процесів квадратні дужки вказують на початку стрілки.

Не рекомендується використовувати без необхідності.

## 7.8. Метод побудови моделей в IDEF0 на основі організаційної структури

Досить часто власники або керівники верхнього рівня ставлять завдання розібратися з поточною ситуацією і з'ясувати доцільність існуючої на підприємстві організаційної структури, чітко визначити функції підрозділів, зрозуміти, хто за них відповідає, з якою ефективністю виконується робота та інше. Це особливо важливо на етапі інформатизації функцій підприємства.

Модель в IDEF0 може допомогти в цьому випадку чітко розписати їх функції, структурувати їх за підрозділами. Побудовану модель можна піддати аналізу і запропонувати зміни як в частині структури підрозділів, так і за складом виконуваних ними функцій.

При такому підході можна будувати модель в IDEF0, спираючись на схему організаційної структури підприємства.

Використовується на початковому етапі побудови БП та визначення ключових центрів відповідальності (фінансових, секретності, безпеки та інших) з метою визначення черговості проведення робіт з розробки ПЗ.

## 7.9. Метод побудови моделей в IDEF0 на основі ланцюжків створення цінності

Якщо розглянути діяльність підприємства з точки зору ланцюжків створення цінності, отримуємо базові моделі діяльності підприємств та організацій, що дуже важливо при створенні та модернізації інформаційної системи. Для цього необхідно виявити ланцюжок послідовно виконуваних бізнес-процесів, на виході якої з'являються продукти чи послуги підприємства.

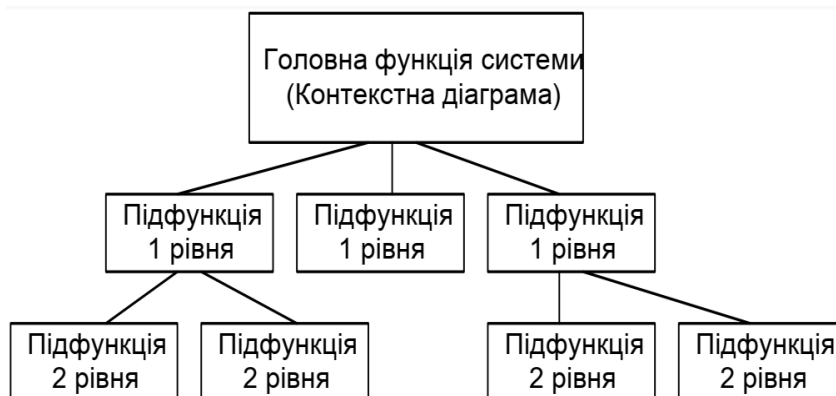


Рис. 7.14 Дерево моделі – зв'язок між моделями (під-функціями)

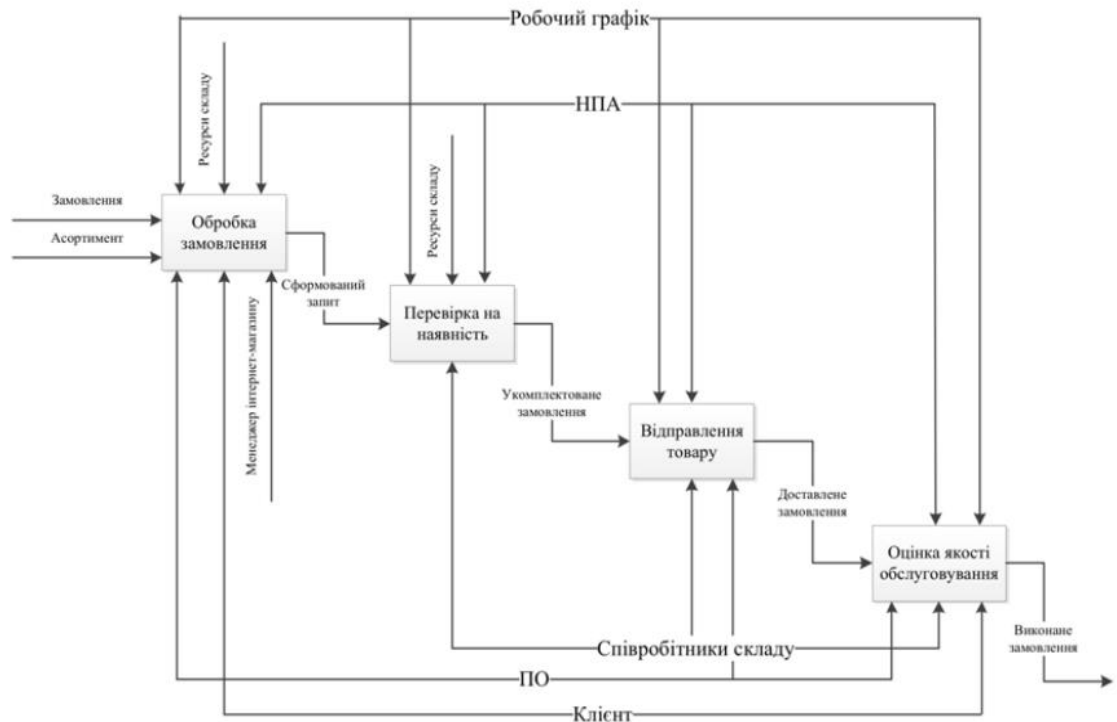


Рис. 7.15 Побудова “As Is”

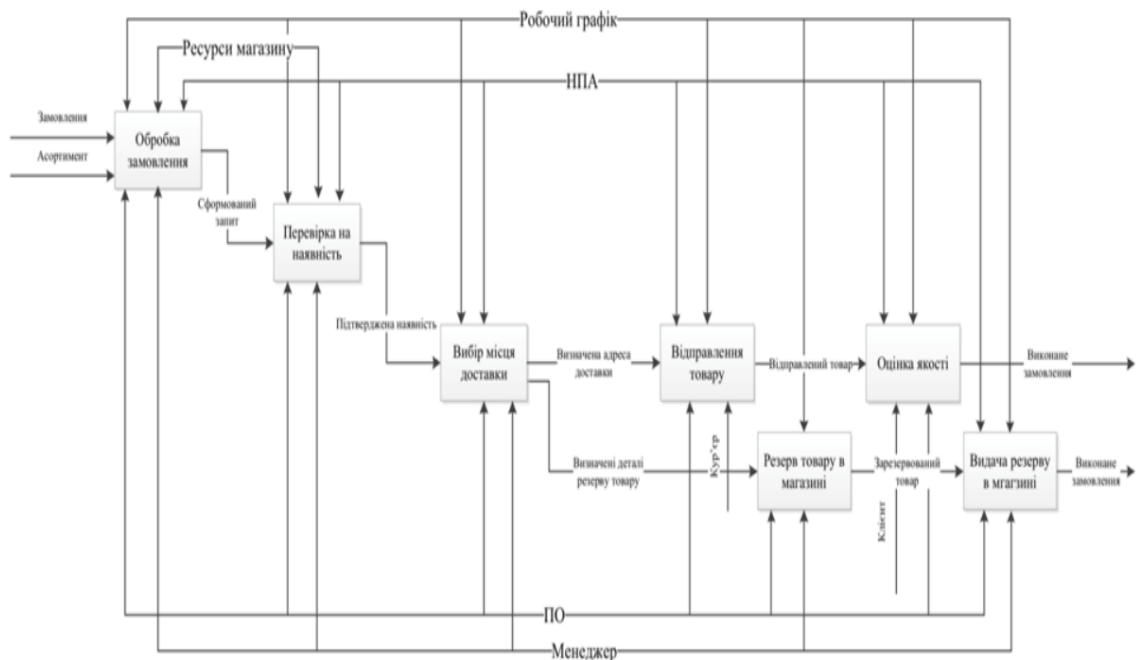


Рис. 7.16 Реінжиніринг в “To Be”

У такий ланцюжок процесів, як правило, включаються не тільки процеси підприємства, а й процеси, що виконуються його зовнішніми контрагентами.

Побудовані на такій основі моделі IDEF0 дають можливість визначити найбільш важливі процеси діяльності підприємства та сценарії дій для створення окремих цінностей (продуктів, послуг, доходів та інше), що з огляду цього мають бути інформатизовані у першу чергу.

Часто використовується на початковому етапі побудови БП для визначення ключових сценаріїв.

Наприклад: ключовий сценарій на хімічному виробництві – викид у атмосферу отруйних хімічних речовин. Цей сценарій підлягає інформатизації серед найперших.

### **Питання до розділу 7**

1. В чому особливість нотації IDEF0?
2. Яка структура моделі IDEF0?
3. Які існують кроки побудови моделі у нотації IDEF0?
4. Які символи та правила використовуються при побудові моделі у нотації IDEF0?
5. Які є правила використання стрілок IDEF0?
6. Що таке розгалуження, злиття, міграція та тунелювання стрілок?
7. В чому суть методу побудови моделей в IDEF0 на основі організаційної структури?
8. В чому суть методу побудови моделей в IDEF0 на основі ланцюжків створення цінності?

## РОЗДІЛ 8. ОПИС ЛОГІКИ СЦЕНАРІЇВ У МЕТОДОЛОГІЇ IDEF3

### 8.1. Нотація IDEF3

Нотація IDEF3 призначена для опису бізнес-процесів нижнього рівня та окремих сценаріїв, послідовність виконання дій яких визначається умовами [6-8].

Особливістю нотації є наявність логічних операторів, за допомогою яких показують альтернативи і місця прийняття рішень в бізнес-процесах.

Основою моделі IDEF3 служить сценарій бізнес-процесу, який виділяє, описує та пов'язує різні можливі чи необхідні послідовності дій або підпроцесів, що аналізуються для наступної інформатизації.

При цьому кожна модель IDEF3 може описувати декілька можливих сценаріїв послідовностей виконання дій.

Сценарій в нотації IDEF3 визначає призначення і межі опису альтернативних шляхів в моделі.

Важливим в нотації IDEF3 є підбір відповідного найменування для позначення дій.

Сценарій повинен бути задокументований.

Зазвичай склад посадових обов'язків ролі є джерелом інформації про процес і може дуже спрощувати побудову моделі, що описується у нотації IDEF3.

Також важливим для системного аналітика є розуміння мети моделювання – набору питань, відповідями на які буде служити модель.

Межі моделювання – тут важливо визначити які частини опису увійдуть, а які не будуть відображені в моделі.

Основні властивості нотації IDEF3:

1. Як правило, використовується для моделювання процесів нижнього рівня.
2. Нотація IDEF3 є доповненням нотації IDEF0 – застосовується для опису декомпозиції блоків процесу.
3. Модель IDEF3 може підрозділятися на окремі блоки та рівні, для конкретизації окремих елементів, що дає можливість декомпозиції.
4. IDEF3 не обмежує жорсткими рамками синтаксису та семантики, що зручно для опису неповних чи не цілісних систем.
5. Опис відбувається за допомогою діаграми.
6. Нотація IDEF3 не має обмежень у кількості складових частин у межах діаграми, ця кількість визначається тільки необхідністю деталізації відповідно до складності конкретного сценарію.

Бізнес-процес у нотації IDEF0 розглядається як сукупність сценаріїв, що виконуються окремими виконавцями чи підрозділами (ролями).

## **8.2. Вимоги IDEF3 до опису бізнес-процесів**

Нотація IDEF3 дозволяє складати схеми навіть дуже складних бізнес-процесів у зрозумілій формі.

Схема IDEF3 демонструє логіку процесу, його учасників та послідовність їх роботи.

Нотація IDEF3 висуває певні вимоги щодо опису бізнес-процесів. Дотримання цих умов зробить процес найбільш інформативним.

Перед створенням бізнес-процесу за допомогою нотації IDEF3 необхідно:

1. визначити необхідні сценарії (опис дій необхідних для досягнення результату);
2. межі опису моделі (область, за яку не варто виходити задля досягнення зручності сприйняття моделі);
3. визначити точку зору (для кого і для чого буде розроблено процес);
4. обрати послідовність та паралельність сценаріїв у описі процесів.

## **8.3. Основні переваги та недоліки методології IDEF3**

Основною відмінністю нотації IDEF3 від IDEF0 є можливість опису логічного ланцюжка дій.

Таким чином, IDEF0 описує "що" робить організація, нотація IDEF3 покликана описати "як" ці дії виконуються.

Переваги нотації IDEF3:

- простота опису та сприйняття — нотація містить невелику кількість елементів;
- обмеження нотації певними рамками (формат А4).

Недоліки нотації IDEF3:

- мінімальна інформативність. Для більш детального опису процесу потрібне використання нотації IDEF3 разом з IDEF0.

## **8.4. Два типи діаграм в IDEF3**

В рамках стандарту IDEF3 виокремлюють два типи стратегій, тобто діаграм, що дозволяють описувати процес з різних точок зору:

1. Діаграма опису послідовності етапів процесу (Process Flow Description Diagrams - PFDD), за допомогою якої моделюється послідовність дій, що реалізуються в рамках бізнес-процесу (процесна модель).

2. Діаграма стану та трансформації об'єкта в процесі (Object State Transition Network - OSTN), за допомогою якої описуються зміни, що відбуваються з об'єктом в ході його обробки.

Що відображає модель IDEF3?

У загальному випадку, процес - це впорядкована послідовність дій.

Процесна модель IDEF3 дозволяє:

- відобразити послідовність дій у процесах;
- показати логіку взаємодії елементів системи.

Мета IDEF3 – дати можливість системним аналітикам і розробникам ПО описати:

- які та як процеси виконуються в певній послідовності;
- які та як дії беруть участь в кожному окремому процесі.

## 8.5. Основні компоненти IDEF3-моделі

Основні компоненти IDEF3-моделі:

1. Одиниці робіт - блоки, дії.
2. Зв'язки – стрілки.
3. Перехрестя – умови.
4. Об'єкти посилань – пояснення.

## 8.6. Одиниці робіт



Рис. 8.1 Блок одиниці робіт

Одиниця робіт (UOW, Unit of Work) (рис. 8.1) є центральним компонентом моделі. Вона визначає важливу з точки зору автоматизації процесу дію, роботу чи групу робіт.

## 8.7. Зв'язки

Зв'язки показують послідовність та взаємовідносини робіт (рис. 8.2). Зв'язки однонаправлені і можуть бути спрямовані куди завгодно. Зазвичай діаграми малюють таким чином, щоб зв'язки були спрямовані зліва направо.

Розрізняють 3 типи зв'язків:

1. старша стрілка (Зв'язок попередження);
2. стрілка відношень (Зв'язок відношення);
3. потік об'єктів (Зв'язок потоків об'єктів).

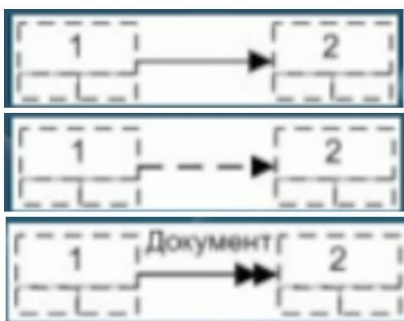


Рис. 8.2 Стрілки зв'язків

## 8.8. Зв'язок «старша стрілка»

Зв'язок типу «тимчасове попередження» - Precedence.

З'єднує одиниці робіт.

Показує, що робота-джерело повинна бути закінчена перш, ніж почеться робота-мета (рис. 8.3).

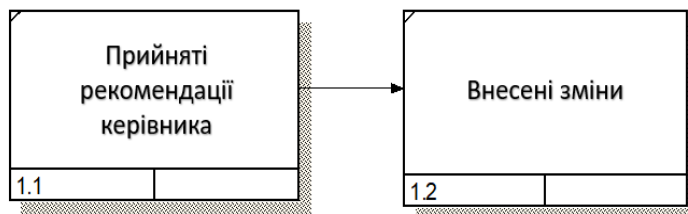


Рис. 8.3 Зв'язок «старша стрілка»

## 8.9. Стрілка відношень

Зв'язок типу нечітке відношення – Relational.

Зображується у вигляді пунктирної лінії, використовується для зображення зв'язку між одиницями робіт, а також між одиницями робіт та об'єктами посилань (частіш) (рис 8.4).

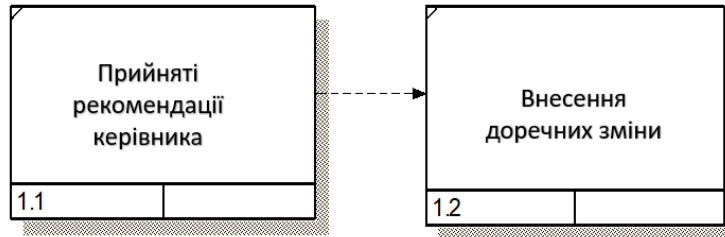


Рис. 8.4 Стрілка співвідношень

## 8.10. Потік об'єктів

Стрілка, що зображає потік об'єктів - Object Flow.

Застосовується для опису того факту, що об'єкт використовується в двох і більш одиницях робіт, наприклад, коли об'єкт породжується в одній роботі і використовується в іншій («рахунок») (рис. 8.5).

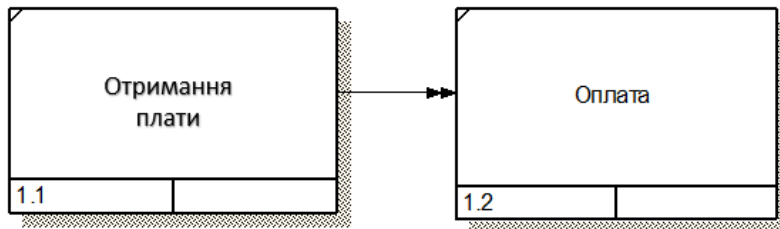





Рис. 8.5 Потік об'єктів

## 8.11. Перехрестя JUNCTION (перехрестя, вузол)

Використовуються для відображення логіки взаємодії стрілок при їх злитті або розгалуженні (таблиця 8.1), для відображення безлічі подій, які можуть або повинні бути завершені перед початком наступної роботи.

Таблиця 8.1 Перехрестя IDEF3

Позначення	Найменування	В разі злиття стрілок (Fan-in Junction)	В разі розгалуження стрілок (Fan-out Junction)
	Асинхронне "І" (Asynchronous AND)	Всі попередні процеси повинні бути завершені	Всі наступні процеси повинні бути запуснені
	Синхронне "І" (Synchronous AND)	Всі попередні процеси завершені одночасно	Всі наступні процеси запускаються одночасно
	Асинхронне "АБО" (Asynchronous OR)	Один або кілька попередніх процесів повинні бути завершені	Один або кілька наступних процесів повинні бути запуснені
	Синхронне "АБО" (Synchronous OR)	Один або кілька попередніх процесів завершені одночасно	Один або кілька наступних процесів запускаються одночасно
	Виключаюче "АБО" XOR (Exclusive OR)	Тільки один попередній процес завершено	Тільки один наступний процес запускається

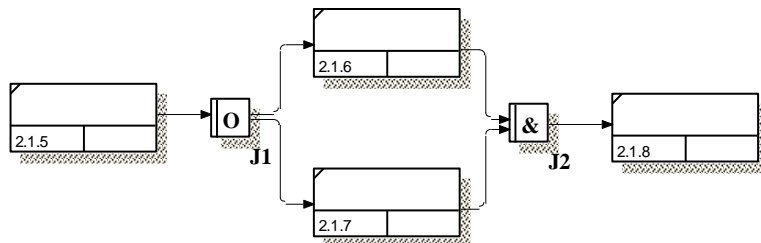


Рис. 8.6 Перехрестя для злиття «І», що слідує за перехрестям для розгалуження, асинхронним «АБО»

Розрізняють перехрестя для злиття і розгалуження стрілок.

Перехрестя не можуть бути одночасно використані для злиття і розгалуження стрілок.

Всі перехрестя на діаграмах нумеруються, кожен номер має префікс J (junction - перехрестя, вузол, точка з'єднання).

На відміну від інших методологій (IDEF0) стрілки можуть зливатися або розгалужуватися тільки через перехрестя.

## 8.12. Правила створення перехресть

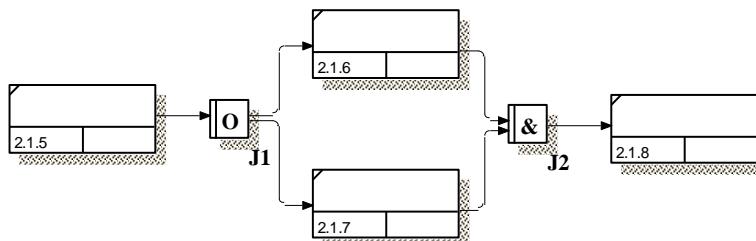


Рис. 8.6 Перехрестя для злиття «І», що слідує за перехрестям для розгалуження, асинхронним «АБО»

Кожному перехрестю для злиття повинне передувати перехрестя для розгалуження (рис. 8.6).

Перехрестя для злиття «І» не може слідувати за перехрестям для розгалуження типу синхронного або асинхронного «АБО».

Перехрестя для злиття «І» не може слідувати за перехрестям типу виняткового «АБО» (рис. 8.7).

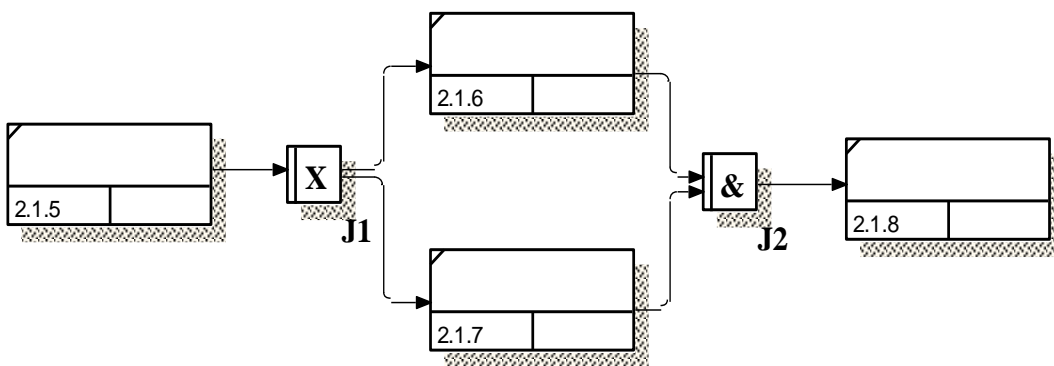


Рис. 8.7 Перехрестя для злиття «І», що слідує за перехрестям розгалуження, винятковим «АБО»

Перехрестя для злиття, типу виняткового «АБО», не може слідувати за перехрестям для розгалуження типу «І» (рис. 8.8).

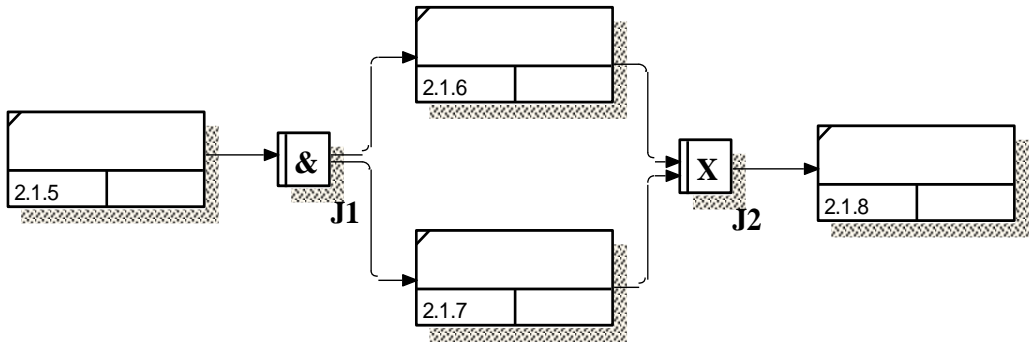


Рис. 8.8 Перехрестя для злиття, типу виняткового «АБО», що слідує за перехрестям розгалуження «І»

Перехрестя, що має одну стрілку на одній стороні, повинно мати більше однієї стрілки на іншій (рис. 8.9, 8.10, 8.11).

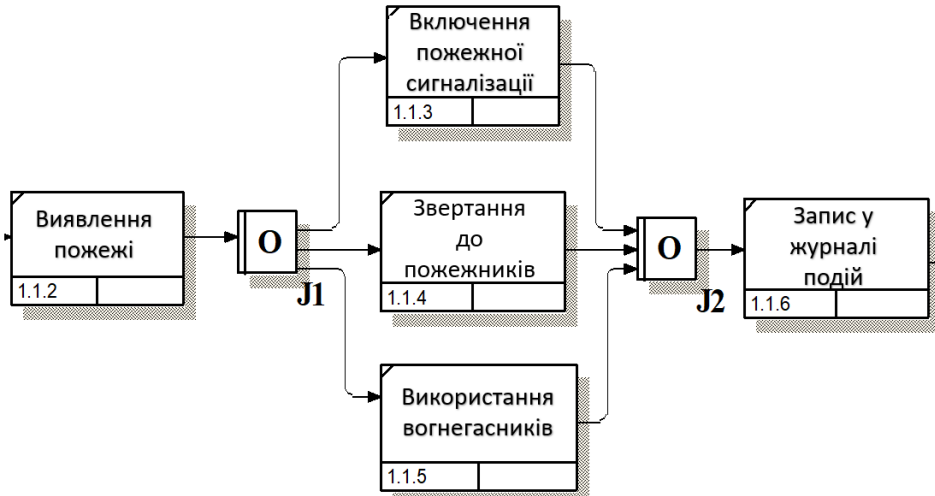


Рис. 8.9 Приклад з перехрестями розгалуження та злиття типу асинхронного «АБО»

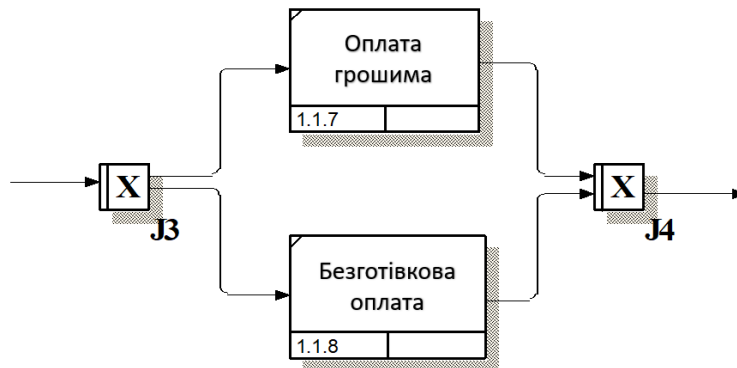


Рис. 8.10 Приклад з перехрестями розгалуження та злиття типу виняткового «АБО»

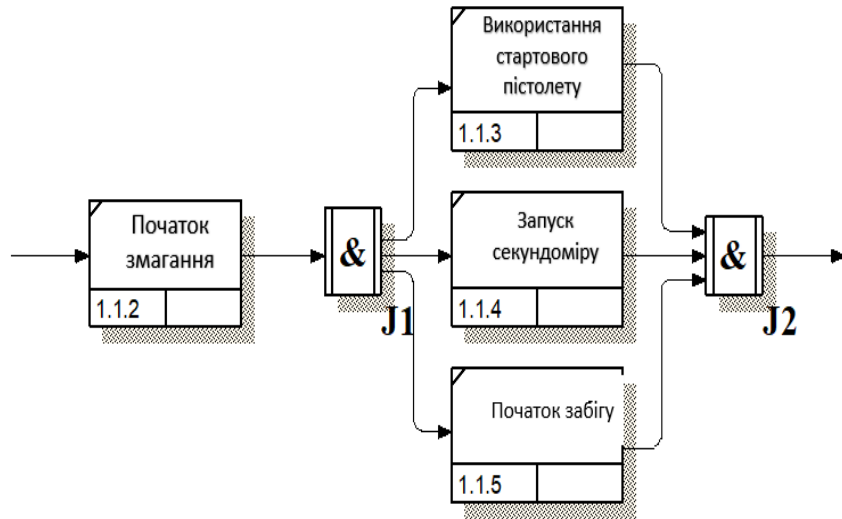


Рис. 8.11 Приклад з перехрестями розгалуження та злиття типу «І»

### 8.13. Комбінації перехресть

Перехрестя можуть комбінуватися для створення складних з'єднань (рис. 8.12).

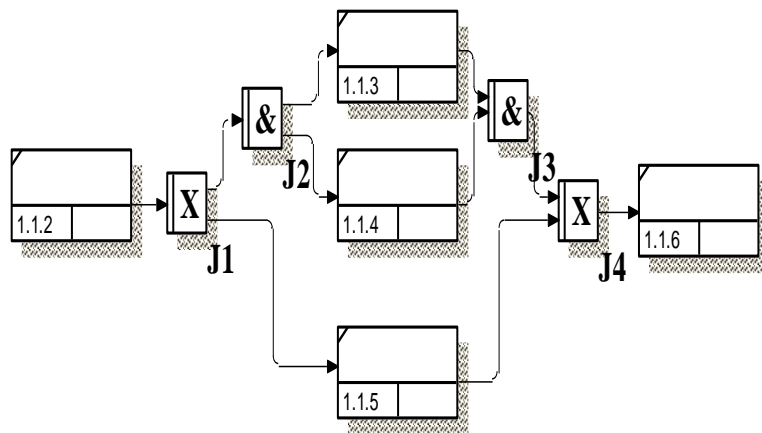


Рис. 8.12 Комбінація перехресть

### 8.14. Об'єкти посилань

Об'єкт посилань:

- висловлює ідею, концепцію даних, які не можна пов'язати зі стрілкою, перехрестям, роботою (рис. 8.13);

– використовується при побудові діаграми для залучення уваги користувача до будь-яких важливих аспектів моделі.

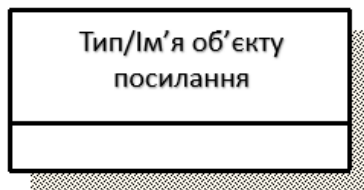


Рис. 8.13 Об'єкти посилань

### 8.15. Типи об'єктів посилань

В таблиці 8.2 перелічено типи об'єктів посилань з описом їх призначення.

Таблиця 8.2 Типи об'єктів посилань

п об'єкту посилань	Призначення
1. Важливий об'єкт чи учасник Object	Використовується для опису того, що в дії бере участь який-небудь важливий об'єкт_(важливий для цього процесу, що впливає чи без кого чи чого неможлива якась дія)
2. Посилання GOTO	Використовується для реалізації циклічності виконуваних дій. Цей об'єкт також може відноситись до перехрестя
3. Одиниця дій UOB (Unit of Behavior)	Використовується для багаторазового відображення на діаграмі одної і тої ж дії, але без циклу
4. Замітка Note	Використовується для документування будь-якої важливої інформації загального характеру, яка відноситься для зображеного на діаграмах.  Служить альтернативою методу розміщення текстових приміток_безпосередньо на діаграмі
5. Уточнення Elaboration ELAB	Для уточнення або більш детального опису зображеного на діаграмі.  Зазвичай використовується для уточнення опису злиття чи розгалуження стрілок на перехрестях

## 8.16. Декомпозиція робіт в IDEF3

У IDEF3 декомпозиція використовується для деталізації робіт.

Методологія IDEF3 дозволяє виконувати декомпозицію роботи багаторазово, тобто робота може мати безліч дочірніх робіт.

Це дозволяє в одній моделі описати альтернативні сценарії.

Можливість множинної декомпозиції надає додаткові вимоги до нумерації робіт.

## 8.17. Нумерація робіт в IDEF3

Номер роботи складається з номера батьківської роботи, версії декомпозиції і власного номера роботи (рис. 8.14, 8.15) на поточній діаграмі.

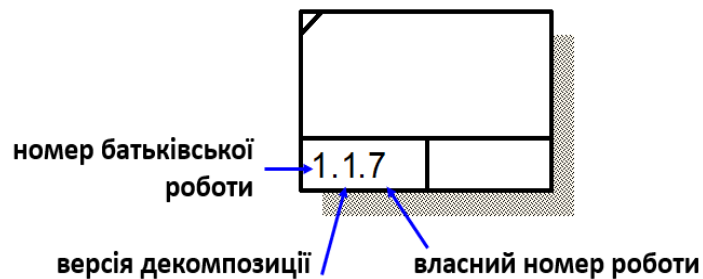


Рис. 8.14 Нумерація блоків-робіт в IDEF3

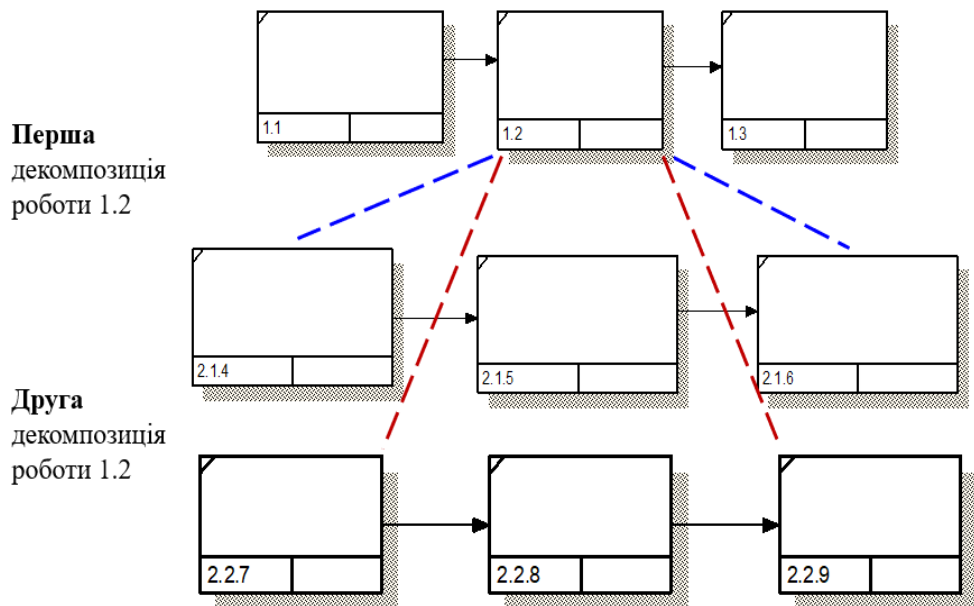


Рис. 8.15 Структура множинної декомпозиції робіт

## 8.18. Приклад побудови моделі IDEF3

Розглянемо на прикладі побудови динамічної моделі процесу «Виконання дипломної роботи».

Почнемо з побудови контекстної діаграми (рис. 8.16).

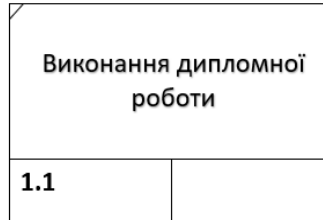


Рис. 8.16 Блок контекстної діаграми

Виконаємо декомпозицію контекстної діаграми (рис. 8.17).

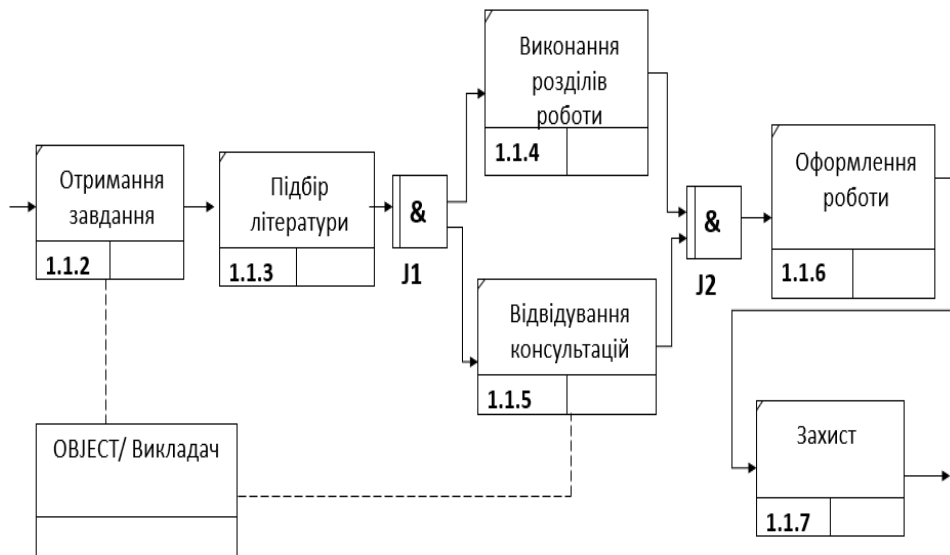


Рис. 8.17 Декомпозиція контекстної діаграми

**Примітка:** Зверніть увагу на нумерацію одиниць робіт. Батьківської є робота з власним номером 1. Вона декомпозується перший раз, отже, версія декомпозиції = 1, далі йде власний номер одиниці робіт в рамках моделі (2-7).

Виконаємо декомпозицію UOW №4 - «Виконання розділів роботи» (рис. 8.18).

Сценарій 1: 8.(9&10).11  
 Сценарій 2: 8.(9&10).(9&10).11

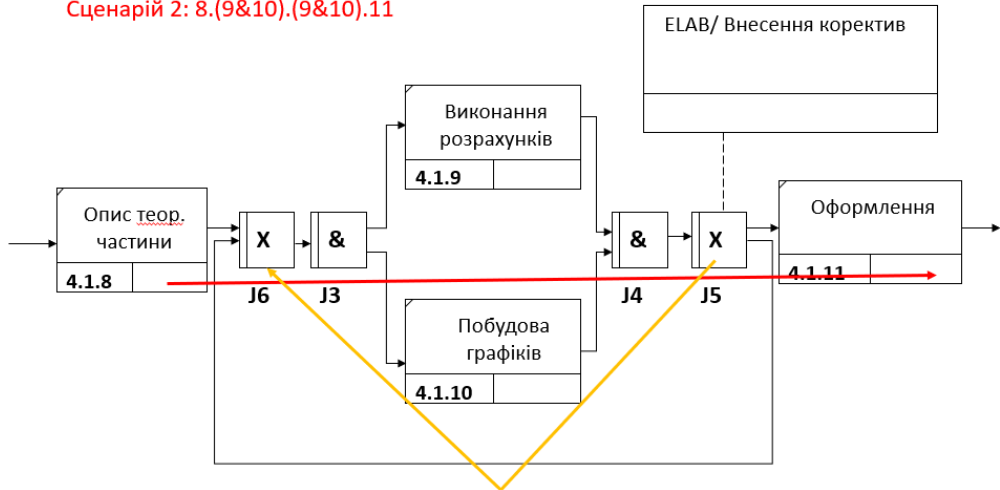


Рис. 8.18 декомпозицію UOW №4 з двома сценаріями

Продекомпуємо повторно контекстну діаграму (у вигляді сценарію IDEF3 для виконання курсової роботи з «Інформатики та програмування») (рис. 8.19, 8.20).

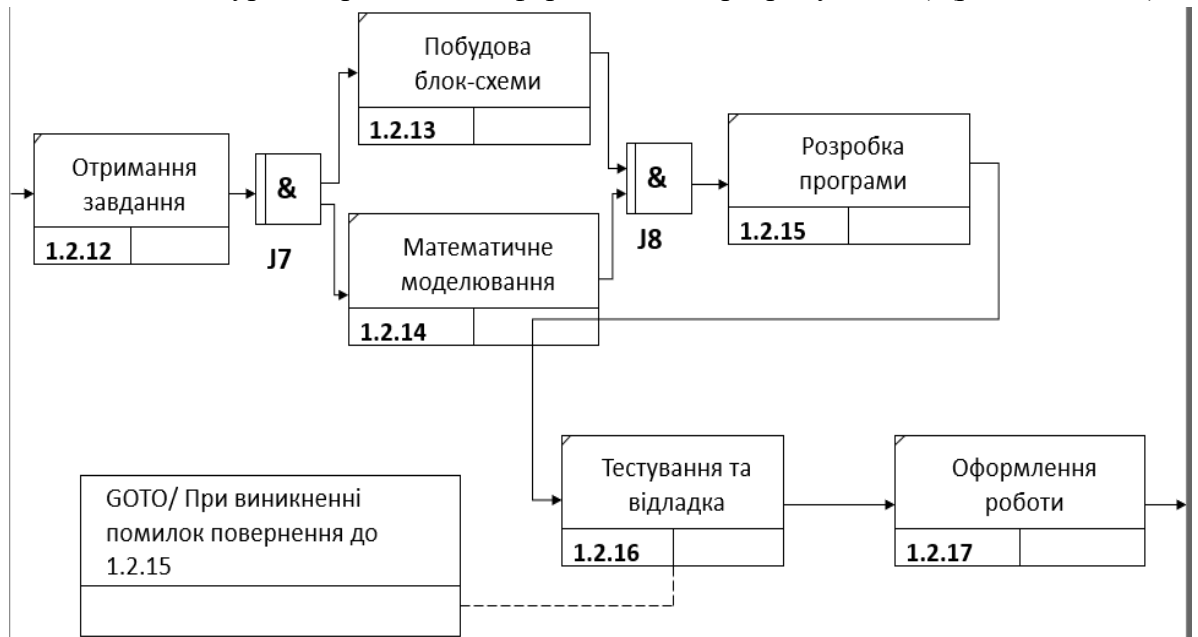
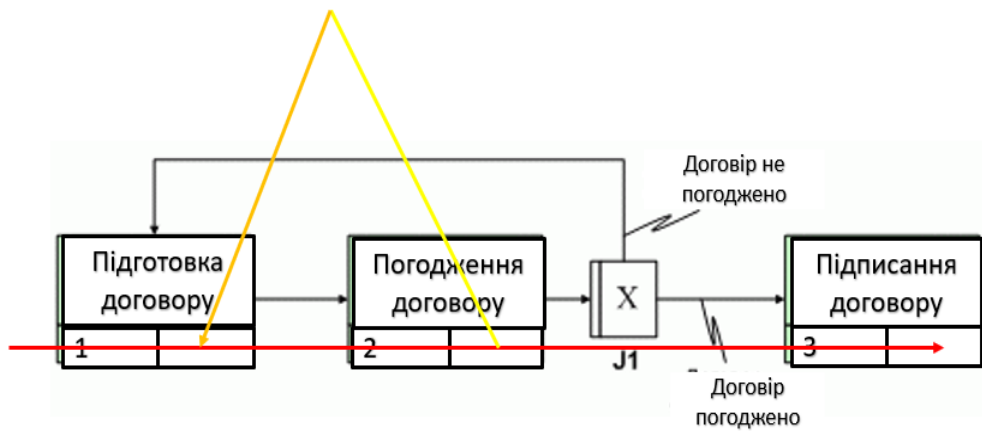


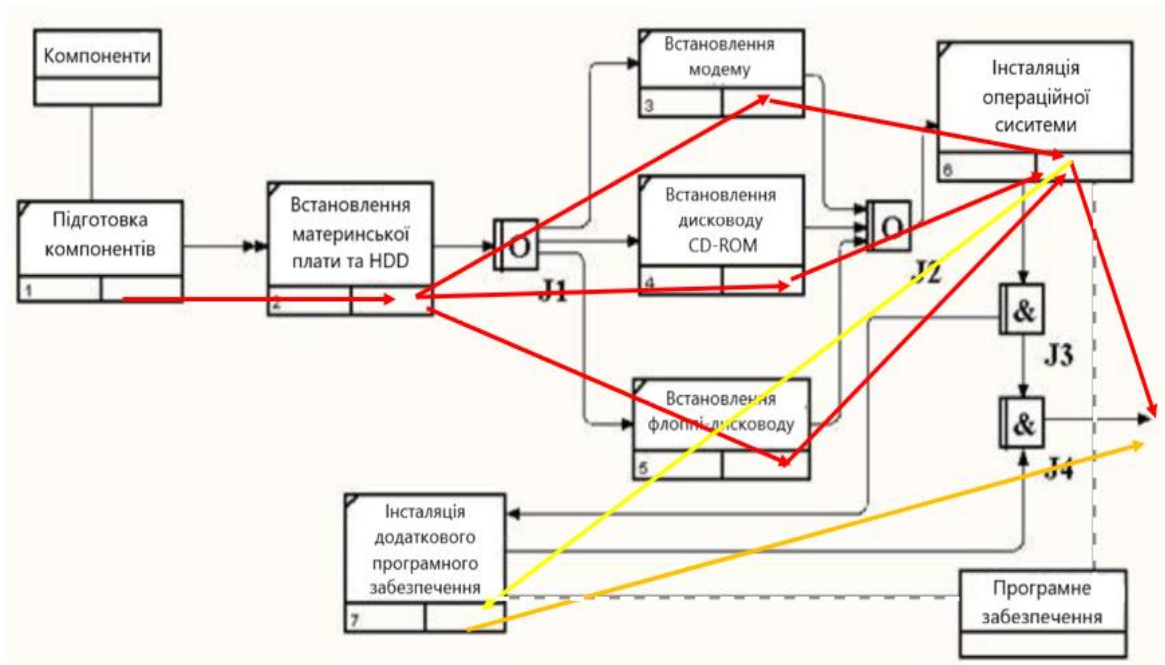
Рис. 8.19 Декомпозиція контекстної діаграми у вигляді сценарію IDEF3



Сценарій 1: 1.2.3.

Сценарій 2: 1.2.1.2.3.

Рис. 8.20 Зворотній зв'язок при невиконанні умов



Сценарій 1-3: 1.2.(3.!4.!5.).6

Сценарій 4-6: 1.2.(3.!4.!5.).6.7.

Рис. 8.21 Збірка комп'ютера

## **Питання до розділу 8**

1. В чому особливість нотації IDEF3?
2. Які властивості та можливості нотації IDEF3?
3. Які вимоги IDEF3 до опису бізнес-процесів?
4. Які основні переваги та недоліки IDEF3?
5. Які є два типи діаграм IDEF3?
6. Які основні компоненти IDEF3?
7. Які основні правила створення перехресть IDEF3?
8. Які є типи об'єктів посилань в IDEF3?
9. В чому особливість нумерації робіт в IDEF3?

## РОЗДІЛ 9. ВИКОРИСТАННЯ CASE-ТЕХНОЛОГІЇ В МОДЕЛЮВАННІ БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ

### 9.1. Методологія об'єктно-орієнтованого аналізу в сценарному аналізі

Необхідність аналізу предметної області, при розробці масштабних проектів, до початку написання програм, була усвідомлена давно.

Предметна область включає в себе тільки ті об'єкти та зв'язки між ними, які необхідні для описання вимог та умов рішення деякої задачі.

Використання методології об'єктно-орієнтованого аналізу та проектування при інформатизації різнорідних об'єктів відбувається при двох факторах, з одного боку, якщо розроблені різні засоби формалізації предметної області, а з іншого – якщо наявні відповідні спеціалісти, які володіють цією методологією та однаково її розуміють.

На даному етапі з'явився новий тип спеціаліста, який отримав назву аналітика та архітектора систем інформатизації.

Об'єктно-орієнтований аналіз і проектування (ООАП, Object-Oriented Analysis/Design) – це технологія розробки програмних систем, в основу яких покладено об'єктно-орієнтована методологія представлення предметної області у вигляді об'єктів, які являються екземплярами відповідних класів.

Об'єктно-орієнтований аналіз – це метод аналізу, що досліджує вимоги до системи з точки зору класів та об'єктів, які відносяться до словника предметної області.

Об'єктно-орієнтоване проектування – це метод проектування, який містить в собі процес об'єктно-орієнтованої декомпозиції і систему позначень для представлення моделі системи, що проектується.

Об'єктно-орієнтоване проектування – це метод розробки програмних систем, оснований на представленні програми у вигляді сукупності взаємодіючих об'єктів, кожний із яких являється екземпляром певного класу, а класи являються членами певної ієрархії наслідування.

Клас (class) – представляє собою сукупність абстракцій реальних об'єктів, які мають загальний набір властивостей та мають однакову поведінку.

Об'єкт (object) – це деяка сутність реального світу або концептуальна сутність з чітко обмеженими межами. Кожний об'єкт в системі має три характеристики: стан, поведінка та індивідуальність. Об'єкт в контексті ООП розглядається, як екземпляр відповідного класу.

Станом (state) об'єкта називають набір умов, в якій він може перебувати. Стан об'єкта характеризується переліком всіх властивостей даного об'єкта і поточними значеннями кожної з цих властивостей – атрибутів (attribute) об'єкта.

Поведінка об'єкта (behavior) – визначає, як об'єкт реагує на запити інших об'єктів, та що може робити сам об'єкт - це його перевірюча діяльність. Поведінка реалізується за допомогою набору операцій (operation) об'єкта.

Індивідуальність (identity) означає, що кожний об'єкт унікальний, навіть якщо його стан ідентичний стану іншого об'єкта. Унікальність об'єкта реалізується в його імені: два різних об'єкта, що належать одному класу не можуть мати однакові імена.

Поділ процесу розробки складних програмних додатків на окремих етапах сприяла постановці концепції життєвого циклу програми. Під життєвим циклом (ЖЦ) програми розуміють сукупність взаємопов'язаних і наступних у часі етапів, починаючи від розробки вимог до неї та закінчуючи повною відмовою від використання програми.

Стандарт ISO 12207, хоч і описує загальну структуру ЖЦ програми, не конкретизує деталі виконання тих чи інших етапів.

ЖЦ програми складається з наступних етапів:

1. Аналіз предметної області та формування вимог до програми.
2. Проектування структури програми.
3. Реалізація програми в кодах (власне саме програмування).
4. Впровадження програми на об'єкті інформатизації.
5. Супроводження програми.
6. Відмова від використання програми.

Методологія ООАП тісно пов'язана з концепцією автоматизованої розробки програмного забезпечення (Computer Aided Software Engineering, CASE).

Графічні нотації, реалізовані в тих чи інших CASE – засобах значно відрізнялися, що визвало появу практичних труднощів в розширенні застосування методології ООАП.

Поява уніфікованої мови програмування (Unified Modelling Language, UML) [1, 3, 5, 6], яка орієнтована на рішення задач перших двох етапів ЖЦ програми (Аналіз предметної області та формування вимог до програми. Проектування структури програми) дало можливість значно спростити та розширити застосування методології ООАП.

Серед всіх розробників CASE-засобів саме компанія Rational Software одна з перших зрозуміла стратегічну перспективність розвитку об'єктно-орієнтованих технологій аналізу та проектування програмних систем. Ця ж компанія першою

розробила інструментальний об'єктно-орієнтований CASE-засіб, в якому була реалізована мова UML, як базова нотація візуального моделювання.

UML (Unified Modeling Language - уніфікована мова моделювання) - мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення.

UML є мовою широкого профілю, це відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи, званої UML моделлю.

UML був створений для визначення, візуалізації, проектування та документування в основному програмних систем.

Використання UML не обмежується моделюванням програмного забезпечення. Його також використовують для моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

UML дозволяє розробникам ПЗ досягти угоди в графічних позначеннях для представлення спільних понять (таких як клас, компонент, узагальнення (generalization), об'єднання (aggregation) і поведінку) і більше сконцентруватися на проектуванні та архітектурі.



Рис. 9.1 Основні види діаграм UML

Головні цілі в розробці UML:

- надати користувачам готову до використання виразну мову візуального моделювання, що дозволяє їм розробляти осмислені моделі та обмінюватися ними;
- передбачити механізми розширення і спеціалізації для розширення базових концепцій;
- забезпечити незалежність від конкретних мов програмування та процесів розробки;

- забезпечити формальну основу для розуміння цієї мови моделювання (мова повинна бути одночасно точною та доступною для розуміння, без зайвого формалізму);
- інтегрувати кращий практичний досвід.

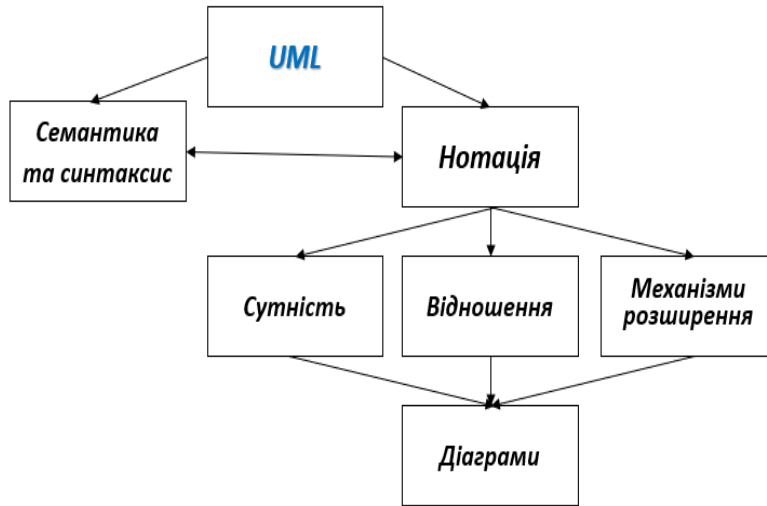


Рис. 9.2 Структура UML

Семантика – розділ мовознавства, що вивчає значення одиниць мови, преш за все її слів та словосполучень.

Синтаксис – спосіб поєднання слів та їх форм в словосполучення та речення, поєднання речень, способи створення висловів, як частини тексту.

Нотація UML є графічною інтерпретацією семантики для її візуального представлення (рис. 9.1, 9.2).

В UML визначено три типи сутностей:

1. Структурні – абстракція, яка є відображенням складу концептуального або фізичного об'єкту;
2. Групуючі – елемент, що використовується для деякого смислового поєднання елементів діаграми.
3. Пояснюючі (анотації) – коментар до елементів діаграми.


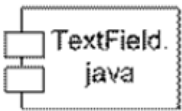
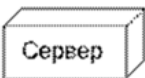
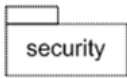
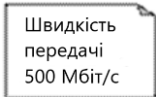
## 9.2. Сутності

В таблиці 9.1 розглянуто сутності нотації UML.

Таблиця 9.1 Сутності

Тип	Найменування	Позначення	Визначення (семантика)
1	2	3	4
Структурна	Клас (class)		Множина об'єктів, що мають загальну структуру
	Об'єкт (object)		Абстракція реальної або вигаданої сутності з чітко вираженими концептуальними границями, індивідуальністю (ідентичністю), станом та поведінкою. З точки зору UML об'єкти являються екземплярами класа (екземплярами сутності)
	Інтерфейс (interface)	 іРозрахунок	Сукупність, що визначає сервіс (набір послуг), що надаються класом або компонентом
	Актор (actor)	 Інженер служби	Зовнішня по відношенню до системи сутність, яка взаємодіє з системою та використовує її функціональні можливості для досягнення певних цілей або рішення часткових задач. Таким чином - актор - це зовнішнє джерело або приймач інформації
	Варіанти використання (use case)	 Розрахунок руху поїзда	Описані послідовності виконуємих системою дій, що призводить до значимого для актора результату
	Стан (state)	 Справний	Описання моменту в ході життя сутності, коли вона задовольняє деякій умові, виконує деяку діяльність або просто чекає настання деякої події

Продовження табл. 9.1

1	2	3	4
Структурна	Кооперація (collaboration)		Описання сукупності екземплярів, об'єктів та їх взаємодії в процесі вирішення деякої задачі
	Компонент (component)		Фізична частина системи (файл, в тому числі модулі системи що забезпечують реалізацію погодженого набору інтерфейсів
	Вузол (node)		Фізична частина системи (комп'ютер, принтер ...), що надає ресурси для задачі
Групуюча	Пакет (packages)		Загальний механізм групування елементів. На відміну від компонента, пакет — чисто концептуальне (абстрактне) поняття
Пояснюю	Примітка (comment)		Коментар до елемента

Для асоціації, агрегації та композиції може вказуватися кратність (англ. Multiplicity), що характеризує загальну кількість екземплярів сутностей, що беруть участь у відношенні. Вона, як правило, вказується з кожного боку відношення біля відповідної сутності.







Кратність може вказуватися наступними способами:

- будь-яка кількість екземплярів, в тому числі й жодного;
- ціле невід'ємне число - кратність строго фіксована та дорівнює зазначеному числу (наприклад: 1, 2 або 5);
- діапазон цілих невід'ємних чисел «перше число .. друге число» (наприклад: 1..5, 2..10 або 0..5);
- діапазон чисел від конкретного початкового значення до довільного кінцевого «перше число .. \*» (наприклад: 1 .. \*, 5 .. \* або 0 .. \*);

– перерахування цілих невід'ємних чисел та діапазонів через кому (наприклад: 1, 3..5, 10, 15 .. \*).

Якщо кратність не вказана, то приймається значення, рівне 1. Кратність екземплярів сутностей, що беруть участь в залежності, узагальненні та реалізації, завжди приймається рівною 1.

Таблиця 9.2 Відношення

Назва	Позначення	Визначення (семантика)
Асоціація (association)		Відношення, яке описує значущий зв'язок між двома і більше сутностями. Найбільш загальний вигляд відносини
Агрегація (aggregation)		Підвид асоціації, який описує зв'язок «частина» - «ціле», в якому «частина» може існувати окремо від «цілого». Ромб вказується з боку «цілого». Ставлення вказується тільки між сутностями одного типу
Композиція (composition)		Підвид агрегації, в якому «частини» не можуть існувати окремо від «цілого». Як правило, «частини» створюються і знищуються одночасно з «цілим»
Залежність (dependency)		Відношення між двома сутностями, при якому зміна в одній сутності (незалежній) може впливати на стан або поведінку іншої сутності (залежної). З боку стрілки вказується незалежна сутність.
Узагальнення (generalization)		Відношення між узагальненою сутністю (предком, батьком) і спеціалізованою сутністю (нащадком). Трикутник вказується з боку батька. Ставлення вказується тільки між сутностями одного типу
Реалізація (realization)		Відношення між сутностями, де одна сутність визначає дію, яку інша сутність зобов'язується виконати. Відносини використовуються в двох випадках: між інтерфейсами та класами (або компонентами), між варіантами використання і коопераціями. З боку стрілки вказується сутність, то визначає дію (інтерфейс або варіант використання)

В таблиці 9.3 перераховано механізми нотації UML

Таблиця 9.3 Механізми розширення

Назва	Позначення	Визначення (семантика)
Стереотип (stereotype)	« »	Позначення, що уточнює семантику елемента нотації (наприклад: залежність зі стереотипом «include» розглядається, як відношення включення, а клас зі стереотипом «boundary» - граничний клас)
Сторожова умова (Сторожовое условие , guard condition)	[ ]	Логічна умова (наприклад: [A > B] або [ідентифікація виконана])
Обмеження (constraint)	{ }	Правило, що обмежує семантику елемента моделі ( наприклад, {час виконання менше 10 мс})
Позначене значення (tagged value)	{ }	Нове або уточнююча властивість елемента нотації (наприклад: {version = 3.2} або {час виконання меншу 10 мс})

Діаграма – це група елементів нотації для відображення деякого аспекту програмної системи для об’єкту інформатизації.

Діаграми являють собою, як правило, зв’язний граф, в якому сутності це вершини, а відносини - дуги.

Діаграми (таблиця 9.4) можуть описувати сценарії роботи програмної системи при вирішенні конкретної задачі.

Таблиця 9.4 Діаграми

Діаграма	Позначення	Тип діаграми (моделі ІС)		
		По ступеню фізичної реалізації	По відображенню динамки	по аспекту, що відображається
Варіанти використання (use case)	Відображає функції системи, взаємодію між акторами та функціями	Логічна	Динамічна	Функціональна
Класів (class)	Відображає набір класів, інтерфейсів і відношення між ними	Логічна або фізична	Статистична	Функціонально-інформаційна

## **Питання до розділу 9**

1. В чому особливість об'єктно-орієнтованого аналізу?
2. Що включає життєвий цикл програм?
3. Які головні цілі розробки UML-діаграм?
4. Які основні види діаграм в UML?
5. Яка структура компонентів UML-діаграм?

## **РОЗДІЛ 10. ОБ'ЄКТНО-ОРІЄНТОВАНИЙ АНАЛІЗ І ПРОЕКТУВАННЯ**

### **10.1. Діаграма Варіантів Використання (Use Case Diagram)**

Розробка діаграми варіантів використання переслідує цілі:

1. Визначити сутності (акторів) та їх можливі варіанти дії, які мають бути враховані при розробці автоматизованої системи.
2. Сформулювати загальні вимоги до функціональної поведінки системи, що проектується.
3. Розробити концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей.
4. Підготувати вихідну документацію для взаємодії розробників системи з її замовниками і користувачами.

### **10.2. Сутність використання Use Case**

Суть даної діаграми складається в наступному: система, що проектується для об'єкту інформатизації представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою так званих варіантів використання [1, 3-10].

При цьому актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на модельовану систему так, як визначить сам розробник.

У свою чергу, варіант використання (use case) служить для опису сервісів, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який використовується системою при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізовано взаємодію акторів з системою.

### **10.3. Варіант використання**

Окремий варіант використання позначається на діаграмі еліпсом, всередині якого міститься його коротка назва або ім'я у формі дієслова з пояснювальними словами (рис. 10.1).

Кожен варіант використання відповідає окремому сервісу, який надає модельовану сутність або систему по запити користувача (актора), тобто визначає спосіб застосування цієї сутності.

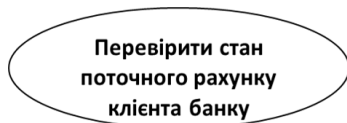


Рис. 10.1 Use Case Diagram

Сервіс, який ініціалізується за запитом користувача, є закінченою послідовністю дій.

## 10.4. Інтерфейс

На діаграмі варіантів використання інтерфейс зображується у вигляді маленького кола, поруч з яким записується його ім'я (рис. 10.2). В якості імені може бути іменник, який характеризує відповідну інформацію або сервіс (наприклад, «форма замовлення», «інформація про клієнта»).



Рис. 10.2 Мале коло відображення інтерфейсу: а – реалізація всіх операцій; б – реалізація тільки необхідного сервісу

Суцільна лінія в цьому випадку вказує на той факт, що пов'язаний з інтерфейсом варіант використання повинен реалізовувати всі операції, необхідні для даного інтерфейсу, а можливо і більше (Рис. 10.2.а). Крім цього, інтерфейси можуть об'єднуватися з варіантами використання пунктирною лінією зі стрілкою (Рис. 10.2.б), яка означає, що варіант використання призначений для специфікації тільки того сервісу, який необхідний для реалізації даного інтерфейсу.

## 10.5. Примітки (notes)

В мові UML примітки призначені для включення в модель довільної текстової інформації, що має безпосереднє відношення до контексту розробленого проекту (рис. 10.3).

Графічно примітки позначаються прямокутником з «загнутим» верхнім правим кутком. У середині прямокутника міститься текст примітки. Примітка може відноситися до будь-якого елементу діаграми, в цьому випадку їх з'єднує пунктирна лінія.

Якщо в примітці зазначається ключове слово «constraint», то дана примітка є обмеженням, що накладається на відповідний елемент моделі, але не на саму діаграму.

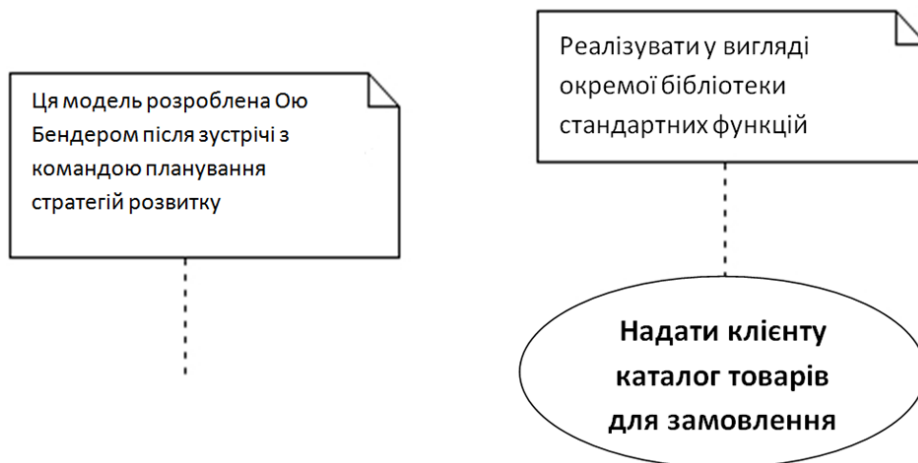


Рис. 10.3 Примітки

## 10.6. Відношення

Між компонентами діаграми варіантів використання можуть існувати різні відносини, які описують взаємодію екземплярів одних акторів і варіантів використання з екземплярами інших акторів і варіантів.

Один актор може взаємодіяти з декількома варіантами використання. В цьому випадку цей актор звертається до кількох сервісів даної системи. У свою чергу один варіант використання може взаємодіяти з декількома акторами, надаючи для всіх них свій сервіс.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

1. відношення асоціації (association);
2. відношення розширення (extend);
3. відношення узагальнення (generalization);
4. відношення включення (include).

## 10.7. Відношення Асоціації

Це відношення встановлює, яку конкретну роль грає актор при взаємодії з екземпляром варіанту використання. На діаграмі варіантів використання, так само як і на інших діаграмах, відношення асоціації позначається суцільною лінією між актором і варіантом використання (рис. 10.4). Ця лінія може мати додаткові умовні позначення, такі, наприклад, як ім'я та кратність (multiplicity).

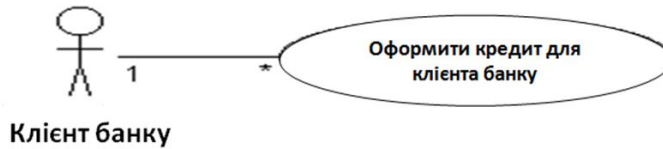


Рис. 10.4 Відношення асоціації

Форми запису кратності:

Ціле невід'ємне число (0999 ...).

Два цілих невід'ємних числа, розділені двома крапками (Гр1..Гр2).

Два символи, розділені двома крапками. При цьому перший з них є цілим невід'ємним числом або 0, а другий - спеціальним символом "\*".

Єдиний символ "\*", який є скороченням записи інтервалу «0 .. \*».

## 10.8. Відношення Розширення

Визначає взаємозв'язок екземплярів окремого варіанту використання з більш загальним варіантом.

Відношення розширення (рис. 10.5) від варіанту використання А до варіанту використання В означає, що властивості екземпляра варіанту використання В можуть бути розширені завдяки наявності додаткових властивостей у варіанту використання А, який доповнює властивості варіанту В.

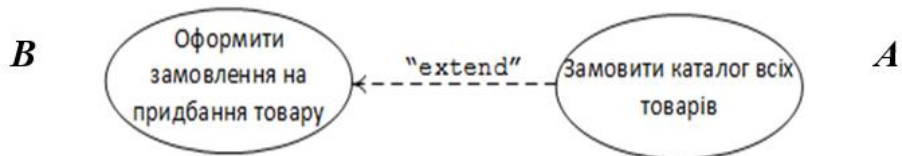


Рис. 10.5 Відношення розширення

Відношення розширення відзначає той факт, що один з варіантів використання може приєднувати до своєї поведінки деяку додаткову поведінку, певне для іншого варіанту використання. Варіант А доповнює варіант В.

Один з варіантів використання може бути розширенням для декількох базових варіантів, а також мати в якості власних розширень кілька інших варіантів. Базовий варіант використання може додатково ніяк не залежати від своїх розширень.

При оформленні замовлення на придбання товару тільки в деяких випадках може знадобитися надання клієнту каталогу всіх товарів. При цьому умовою розширення є запит від клієнта на отримання каталогу товарів. Очевидно, що після отримання каталогу клієнту потрібен певний час на його вивчення, протягом якого оформлення замовлення призупиняється. Після ознайомлення з каталогом клієнт вирішує або на користь вибору окремого товару, або на користь відмови від покупки взагалі.

Сервіс або варіант використання «Оформити замовлення на придбання товару» може відреагувати на вибір клієнта вже після того, як клієнт отримав для ознайомлення каталог товарів (рис. 10.6).

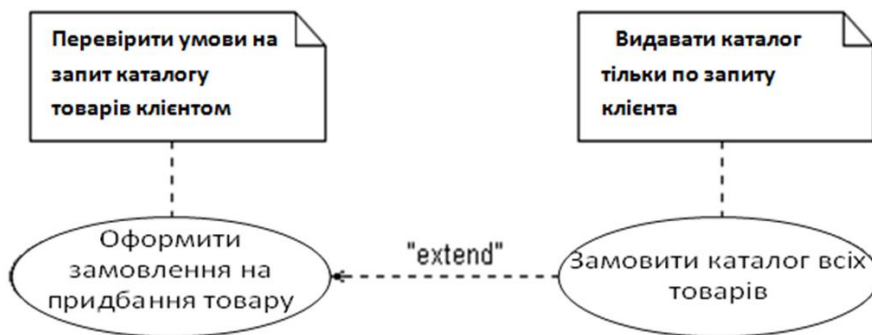


Рис. 10.6 Відношення розширення. Оформлення замовлення на придбання товару

### 10.9. Відношення Узагальнення

Відношення узагальнення (рис. 10.7) служить для вказівки того факту, що деякий варіант використання А може бути узагальненим до варіанту використання В. В цьому випадку варіант А буде спеціалізацією варіанту В. При цьому В називається предком по відношенню А, а варіант А називається нащадком по відношенню до варіанту використання В. Слід підкреслити, що нащадок успадкує всі властивості і поведінку свого батька, а також може бути доповнений новими властивостями і особливостями поведінки.



Рис. 10.7 Відношення узагальнення

Графічно дане відношення позначається суцільною лінією зі стрілкою в формі не зафарбованого трикутника, яка вказує на батьківський варіант використання.

Між окремими акторами також може існувати відношення узагальнення (рис. 10.8). Дане відношення є спрямованим і вказує на факт спеціалізації одних акторів відносно інших.

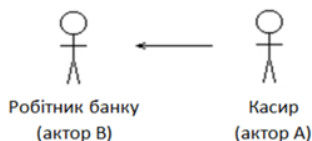


Рис. 10.8 Відношення узагальнення між окремими акторами

### 10.11. Відношення Включення

Відношення включення (рис. 10.9) між двома варіантами використання вказує, що деяка задана поведінка для одного варіанта використання включається як складовий компонент в поведінці іншого. Відношення Включення є спрямованим бінарним відношенням в тому сенсі, що пара екземплярів варіантів використання завжди впорядкована відносно включення (рис. 10.10).

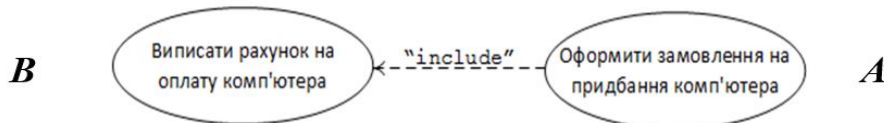


Рис. 10.9 Відношення включення



Рис. 10.10 Приклад. Система продажу товарів по каталогу

Один варіант використання може бути включений в кілька інших варіантів, а також включати в себе інші варіанти.

Відношення включення, спрямоване від варіанту використання А до варіанту використання В, вказує, що кожен екземпляр варіанту А включає в себе функціональні властивості варіанту В.

Графічно дане відношення позначається пунктирною лінією зі стрілкою (варіант відношення залежності), спрямованою від базового варіанту використання до такого, що включається. При цьому дана лінія зі стрілкою позначається ключовим словом «include» («включає»).

Для більш точного визначення можливих дій (функцій) кожного з акторів (ролей) використовується виділення зон відповідальності акторів, які показують можливі сценарії дій цих акторів (рис. 10.11).

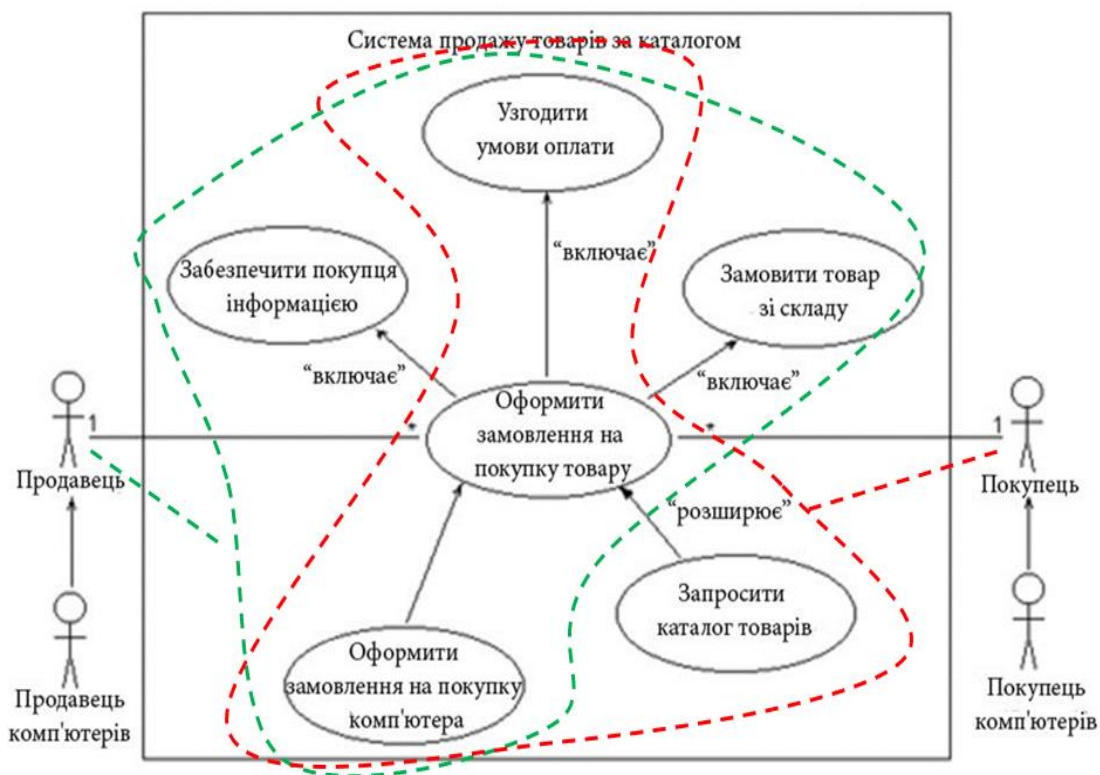


Рис. 10.11 Приклад зони відповідальності

## Питання до розділу 10

1. Яке призначення діаграм варіантів використання (Use Case Diagram)?
2. В чому особливості діаграм варіантів використання?
3. Які основні компоненти Use Case Diagram?

4. Яким чином описуються варіанти використання?
5. Що визначають актори у Use Case Diagram?
6. Яке призначення інтерфейсів та приміток у Use Case Diagram?
7. Які типи відношень у Use Case Diagram?

# РОЗДІЛ 11. МОДЕЛЮВАННЯ СТАТИЧНОЇ СТРУКТУРИ СИСТЕМИ

## 11.1. Діаграма класів (Class Diagram)

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування .

Діаграма класів може показувати різні взаємозв'язки між окремими сутностями предметної області (таблиця 11.1), такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи відносин [1-10, 22].

Діаграма класів є графом, вершинами якого є елементи, які пов'язані різними типами структурних відносин.

Діаграма класів може містити інтерфейси, пакети, відносини і навіть окремі екземпляри, такі як об'єкти і зв'язки.

Коли говорять про подану діаграму, мають на увазі статичну структурну модель проектованої системи. Тому діаграму класів прийнято вважати графічним представленим таких структурних взаємозв'язків логічної моделі системи, що не залежать від часу.

## 11.2. Клас

Клас (class) в мові UML служить для позначення об'єктів, які мають однакову структуру, поведінку і відносини з об'єктами інших класів.



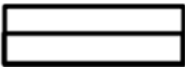
Графічно клас зображується у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на розділи або секції.

У цих розділах можуть зазначатися ім'я класу, атрибути (змінні) і операції (методи) (рис. 11.1).



Рис. 11.1 Діаграми класу: а – ім'я класу; б – ім'я класу з доповнюючим атрибутом; в – ім'я з атрибутом та операцією

Таблиця 11.1 Сутності

Найменування	Позначення	Визначення (семантика)
Клас (Class)		Безліч об'єктів, що мають загальну структуру і поведінку
Об'єкт (Object)		Абстракція реальної чи уявної сутності з чітко вираженими концептуальними межами, індивідуальністю (ідентичністю), станом і поведінкою. З точки зору UML об'єкти є екземплярами класу (екземплярами сутності)
Інтерфейси (Interface)		Набір операцій, що визначає сервіс (набір послуг), що надається класом або компонентом

На початку розробки діаграми класи можуть позначатися прямокутником зі зазначенням тільки імені класу (рис. 11.1.а). А при уточненні та доробці компонентів діаграми опису класів у описі доповнюються атрибути (рис. 11.1.б) і операції (рис. 11.1.в).

Навіть якщо секція атрибутів чи операцій не потрібна, в позначенні класу вона виділяється горизонтальною лінією, щоб відразу відрізнити клас від інших елементів мови UML.

У першому випадку для класу «Прямокутник» (рис. 11.2.а) вказані тільки атрибути - точки на координатній площині, які визначають його розташування.

Для класу «Вікно» (рис. 11.2.б) вказані тільки операції, секція атрибутів залишена порожньою.

Для класу «Рахунок» (рис. 11.2.в) додатково зображена четверта секція, в якій зазначено виняток - відмова від обробки простроченої кредитної картки.



Рис. 11.2 Діаграми класу: а – клас «Прямокутник» з атрибутами; б - клас «Вікно» з операціями; в – клас «Рахунок» з відображенням винятку

### 11.3. Ім'я класу

Ім'я класу має бути унікальним одного опису, що описує сукупністю діаграм класів.

Воно вказується в першій верхній секції прямокутника. На додаток до загального правила найменування елементів мови UML, ім'я класу записується по центру секції імені напівжирним шрифтом і має починатися з великої літери.

Для позначення імена класів використовуються іменники.

Прикладами імен класів можуть бути такі іменники, як «Співробітник», «Компанія», «Керівник», «Клієнт», «Продавець», «Менеджер», «Офіс».

У деяких випадках необхідно явно вказати, до якого пакету належить той чи інший клас.

Для цієї мети використовується спеціальний символ роздільник - подвійна двокрапка «::». Синтаксис рядка імені класу в цьому випадку буде наступний <Ім'я\_Пакету> :: <ім'я\_класу>. Іншими словами, перед ім'ям класу має бути явно вказане ім'я пакету, до якого його слід віднести.

Наприклад, якщо визначено пакет з ім'ям «Банк», то клас «Рахунок» в цьому банку може бути записаний у вигляді: «Банк :: Рахунок».

### 11.4. Атрибути Класу

У другій секції зверху прямокутника класу записуються його атрибути (attributes) або властивості.

Кожному атрибуту класу відповідає окремий рядок тексту, яка складається з квантора видимості атрибута, імені атрибута, його кратності, типу значень атрибута і його вихідного значення (не завжди) : <Квантор видимості> <ім'я атрибута> [кратність] : <Тип атрибута> = <початкове значення> {рядок-властивість}

Квантор видимості може приймати одне з трьох можливих значень.

Символ "+" позначає атрибут з областю видимості типу загальнодоступний (public). Атрибут з цією областю видимості доступний або видимий з будь-якого іншого класу пакета, в якому визначена діаграма.

Символ "#" позначає атрибут з областю видимості типу захищений (protected). Атрибут із цією областю видимості недоступний або не видимий для всіх класів, за винятком підкласів даного класу.

Символ "-" позначає атрибут з областю видимості типу закритий (private). Атрибут із цією областю видимості недоступний або не видимий для всіх класів без виключення.

Квантор видимості можна не вказувати.

Замість умовних графічних позначень можна записувати відповідне ключове слово: public, protected, private.

Ім'я атрибута є рядком тексту, який є ідентифікатором відповідного атрибута і повинен бути унікальним в межах даного класу.

Ім'я атрибута є єдиним обов'язковим елементом атрибута.

Кратність атрибута вказує кількість атрибутів даного типу, що входять до складу класу.

Кратність записується в формі рядка тексту в квадратних дужках після імені відповідного атрибута: [нижня\_границя1 .. верхня\_границя1, нижня\_границя2 .. верхня\_границя2, ..., нижня\_границяк .. верхня\_границяк] , де нижня\_границя і верхня\_границя є позитивними цілими числами.

В якості верхньої межі може використовуватися спеціальний символ "\*", який означає довільне натуральне число.

Якщо кратність атрибута не вказана, то за замовчуванням приймається її значення рівне 1..1.

Приклади завдання імен і типів атрибутів класів:

1. Колір: COLOR - тут колір є ім'ям атрибута, Color-типу даного атрибута. Зазначений запис може визначати традиційно використовувану RGB-модель (червоний, зелений, синій).

2. Ім'я\_співробітника [1..2]: String - тут ім'я\_співробітника є ім'ям атрибута, який служить для представлення інформації про ім'я, а можливо, і по батькові конкретного співробітника. Тип атрибута String (Рядок) якраз і вказує на той факт, що окреме значення імені являє собою рядок тексту з одного або двох слів (наприклад, «Кирило» або «Дмитро Іванович»).

3. Видимість: Boolean - тут видимість є ім'ям абстрактного атрибута, який характеризує наявність візуального представлення відповідного класу на екрані монітора. У цьому випадку тип Boolean означає, що можливими значеннями даного атрибута є одне з двох логічних значень: істина (true) або брехня (false). При цьому значення істина може відповідати наявності графічного зображення на екрані монітора, а значення брехня - його відсутності, про що додатково вказується в пояснювальній тексті.

4. Форма: Багатокутник - тут ім'я атрибута – форма. Може характеризувати такий клас, який є геометричною фігурою на площині. У цьому випадку тип атрибута Багатокутник вказує на той факт, що окрема геометрична фігура може мати форму

трикутника, прямокутника, ромба, п'ятикутника і будь-якого іншого багатокутника, але не окружності або еліпса.

Початкове значення служить для завдання деякого початкового значення для відповідного атрибута в момент створення окремого екземпляра класу.

Приклади початкових значень атрибутів для вищеописаних варіантів завдань атрибутів:

1. Колір: COLOR = (255, 0, 0) - в RGB-моделі кольору це відповідає чистому червоному кольору в якості початкового значення для даного атрибута;

2. Імя\_співробітника [1..2]: String = Іван Іванов - можливо, це перша людина в списку співробітників;

3. Видимість: Boolean = істина - може відповідати ситуації, коли в момент створення екземпляра класу створюється видиме на екрані монітора вікно, відповідне даному об'єкту;

4. Форма: Багатокутник = прямокутник – вказує на первинний вид прямокутника, що розглядається.

Рядок-властивість служить для вказівки значень атрибута, які не можуть бути змінені в програмі при роботі з даним типом об'єктів.

Фігурні дужки позначають фіксоване значення відповідного атрибута для класу в цілому, яке повинні приймати всі новостворювані екземпляри класу без винятку.

Наприклад, рядок-властивість в запису атрибута заробітна\_плата: Currency = {\$ 500} може служити для визначення фіксованої заробітної плати для кожного об'єкта класу «Співробітник» певної посади в деякій організації.

З іншого боку, запис даного атрибута у вигляді заробітна\_плата: Currency = \$ 500 означає вже щось інше, а саме - при створенні нового екземпляра «Співробітник» (аналогія - прийом на роботу нового співробітника) для нього встановлюється за замовчуванням (початкова) заробітна плата в \$ 500.

## 11.5. Операція

У третій зверху секції прямокутника записуються операції або методи класу.

Операція (operation) являє собою сервіс, що надає кожний екземпляр класу за певними умовами.

Кожній операції класу відповідає окремий рядок, який складається з квантора видимості операції, імені операції, вирази типу значення, що повертається операцією, та можливо, рядок-властивість даної операції:

<Квантор видимості> <ім'я операції> (список параметрів) : <Вираз типу значення, що повертається> {рядок-властивість}

Квантор видимості, як і в випадку атрибутів класу, може приймати одне з трьох можливих значень.

Символ "+" позначає операцію з областю видимості типу загальнодоступний (public).

Символ "#" позначає операцію з областю видимості типу захищений (protected).

Символ "-" використовується для позначення операції з областю видимості типу закритий (private). Квантор видимості для операції може бути опущений.

Список параметрів є переліком розділених комою формальних параметрів, кожен з яких може бути представлений в наступному вигляді:

<Вид параметра> <ім'я параметра>: <вираз типу> = <значення параметра за замовчуванням>.

Тут вид параметра – є одне з ключових слів in, out або inout зі значенням in за замовчуванням, в разі якщо вид параметра не вказується.

Ім'я параметра є ідентифікатор відповідного формального параметра.

Вираз типу є залежною від конкретної мови програмування специфікацією типу значення, що повертається для відповідного формального параметра.

Значення за замовчуванням в загальному випадку являє собою вираз для значення формального параметра, синтаксис якого залежить від конкретної мови програмування і підпорядковується прийнятим в ньому обмеженням.

Приклади запису операцій:

1. + створити () - може позначати абстрактну операцію по створенню окремого об'єкта класу, яка є загальнодоступною і не містить формальних параметрів. Ця операція не повертає ніякого значення.

2. + намалювати (форма: Багатокутник = прямокутник, колір\_заливки: Color = (O, O, 255)) - може позначати операцію по зображенню на екрані монітора прямокутної області синього кольору, якщо не вказуються інші значення в якості аргументів даної операції.

3. Запросити\_рахунок\_клієнта (номер\_рахунку: integer): Currency - позначає операцію по встановленню наявності коштів на поточному рахунку клієнта банку. При цьому аргументом даної операції є номер рахунку клієнта, який записується у вигляді цілого числа (наприклад, «123456»). Результатом виконання цієї операції є деяке число, записане в прийнятому грошовому форматі (наприклад, \$ 1,500.00).

4. Видати\_повідомлення (): {"Помилка поділу на нуль"} - сенс даної операції не вимагає пояснень, оскільки міститься в рядку-властивості операції. Дане

повідомлення може з'явитися на екрані монітора в разі спроби поділу деякого числа на нуль, що є неприпустимим.

## 11.6. Відношення Між Класами

Крім внутрішнього устрою або структури класів на відповідній діаграмі вказуються різні відношення між класами (рис. 11.3). При цьому сукупність типів таких відношень фіксована в мові UML і зумовлена семантикою цих типів відношень.

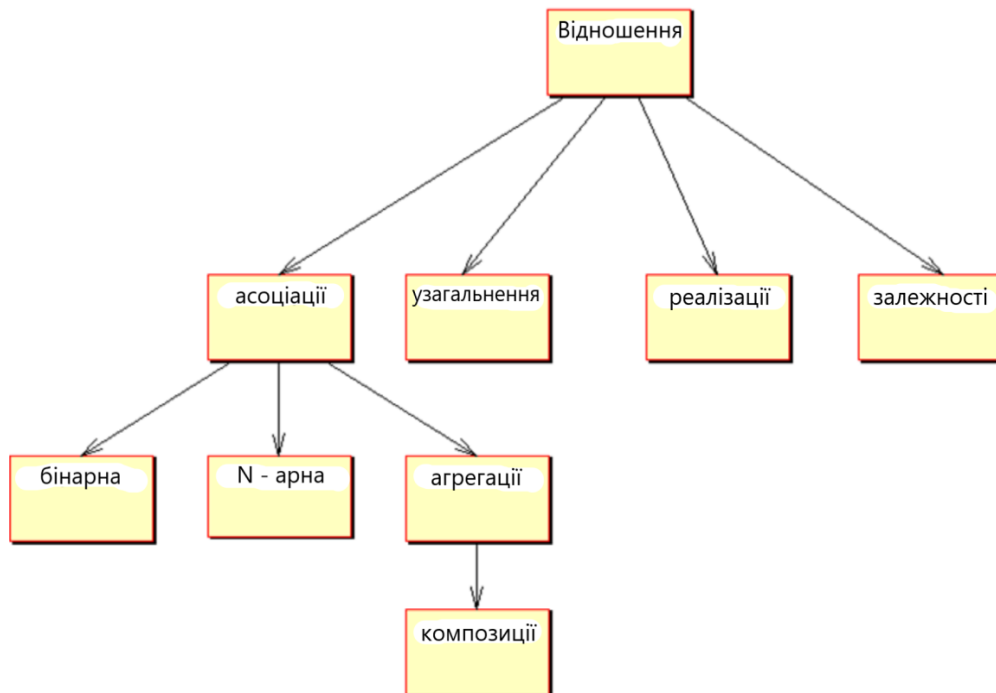


Рис. 11.3 Базові відношення в мові UML

Базовими відношеннями або зв'язками в мові UML є:

1. відношення залежності (dependency relationship);
2. відношення асоціації (association relationship);
3. відношення узагальнення (generalization relationship);
4. відношення реалізації (realization relationship).

Кожне з цих відносин має власне графічне представлення на діаграмі, яке відображає взаємозв'язки між об'єктами відповідних класів (таблиця 9.2).

## 11.7. Відношення Залежності

Відношення залежності (рис. 11.4) використовується в такій ситуації, коли деяка зміна одного елемента моделі може зажадати зміни іншого залежного від нього елемента моделі.

Відношення залежності графічно зображується пунктирною лінією між відповідними елементами зі стрілкою на одному з її кінців («->» або «<-»).

На діаграмі класів дане відношення зв'язує окремі класи між собою, при цьому стрілка спрямована від класу-клієнта залежності до незалежного класу або класу-джерела). На малюнку зображено два класи: Клас\_А і Клас\_Б, при цьому Клас\_Б є джерелом деякої залежності, а Клас\_А - клієнтом цієї залежності.

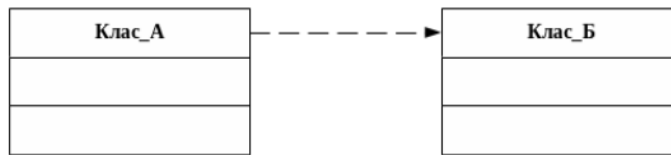


Рис. 11.4 Відношення залежності

Як для класу-клієнта і класу-джерела залежно можуть виступати цілі безлічі елементів моделі. У цьому випадку одна лінія зі стрілкою, що виходить від джерела залежності, розщеплюється в деякій точці на кілька окремих ліній, кожна з яких має окрему стрілку для класу-клієнта.

Наприклад, якщо функціонування Класу\_С залежить від особливостей реалізації Класу\_А і Класу\_Б, то дана залежність може бути зображена наступним чином (рис. 11.5).

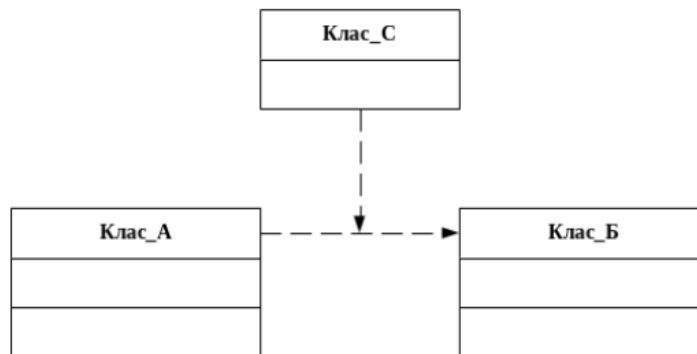


Рис. 11.5 Клас С, що залежить від класі А та Б

## 11.8. Відношення Асоціації

Відношення асоціації відповідає наявності деяких відносин між класами.

Дане відношення позначається суцільною лінією з додатковими спеціальними символами, які характеризують окремі властивості конкретної асоціації.

Найбільш простий випадок даного відношення - бінарна асоціація. Вона пов'язує два класи.

Для бінарної асоціації на діаграмі може бути вказаний порядок проходження класів з використанням трикутника у формі стрілки поруч з ім'ям даної асоціації.

Напрямок стрілки вказує на порядок класів, один з яких є першим (з боку трикутника), а інший - другим (з боку вершини трикутника).

Може вказувати напрямки розвитку використання сутностей у сценаріях.

Приклад відношення бінарної асоціації (рис. 11.6) розглянемо відношення між двома класами - класом «Компанія» і класом «Співробітник». Вони пов'язані між собою бінарної асоціацією.

Робота, ім'я якої позначено на малюнку поруч з лінією асоціації. Для даного відношення визначено порядок проходження класів, першим з яких є клас «Співробітник», а другим - клас «Компанія».

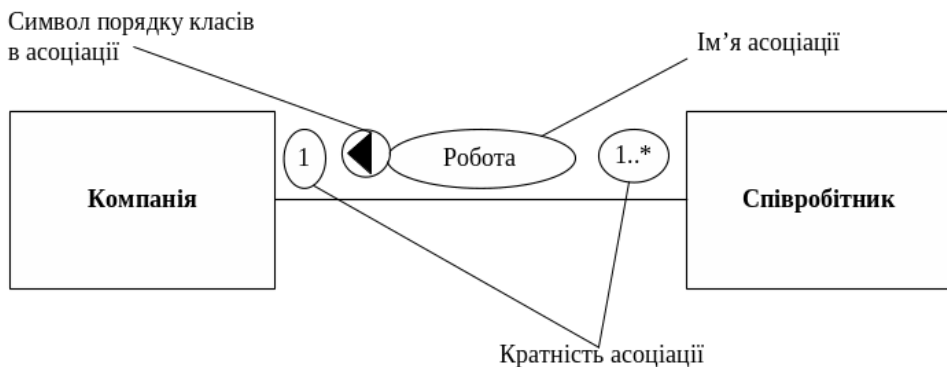


Рис. 11.6 Відношення бінарної асоціації

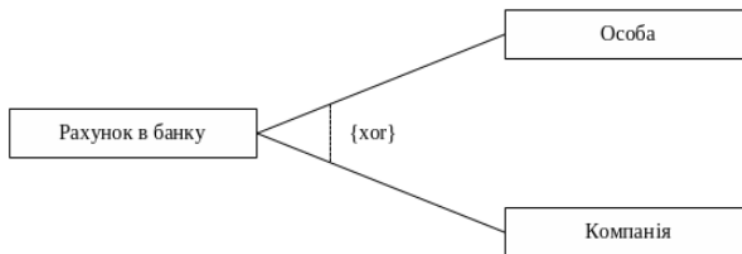


Рис. 11.7 Виключна асоціація (Xor-association)

Окремим випадком відносини асоціації є так звана виключна асоціація (Xor-association) (рис. 11.7, 11.8, 11.9). Семантика даної асоціації вказує на той факт, що з кількох потенційно можливих варіантів даної асоціації в кожен момент часу може використовуватися тільки один її примірник.

На діаграмі класів виключна асоціація зображується пунктирною лінією, що з'єднує дві і більше асоціації, поруч з якою записується рядок-обмеження «{xor}». Наприклад, рахунок в банку може бути відкритий для клієнта, в якості якого може виступати фізична особа або компанія, що зображується за допомогою виключної асоціації.

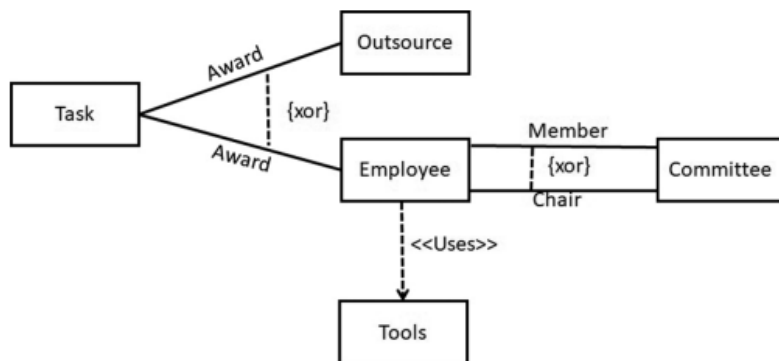


Рис. 11.8 Приклад застосування Xor-association

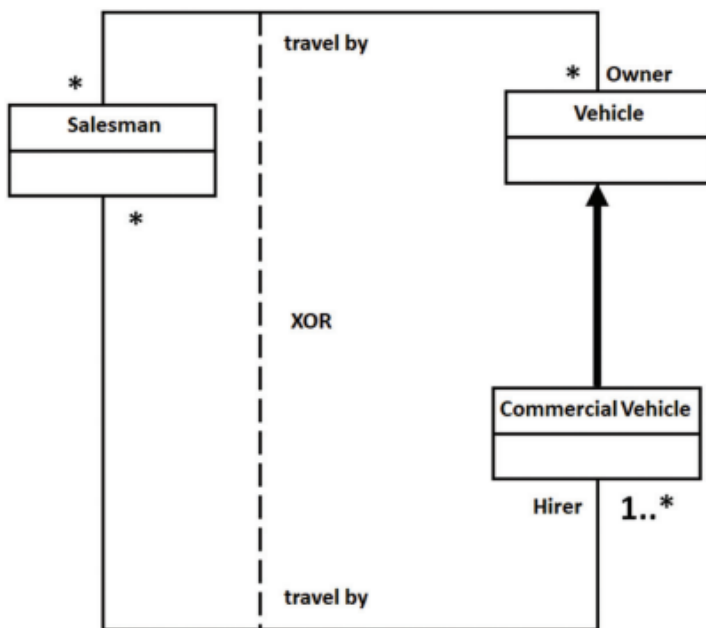
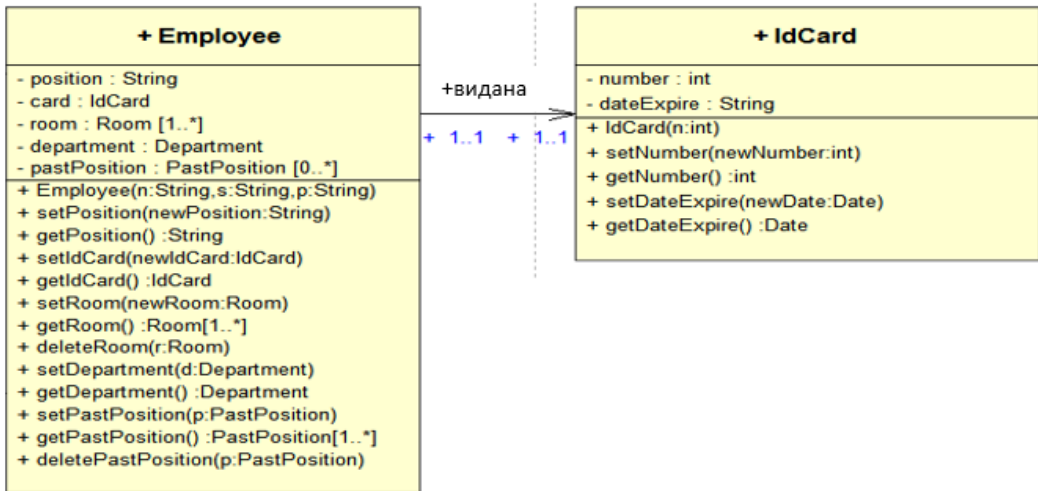


Рис. 11.9 Виключна асоціація (Xor-association)



### Бінарна асоціація

Рис. 11.10 Додавання класу «IdCard» у модель

### N-арна асоціація

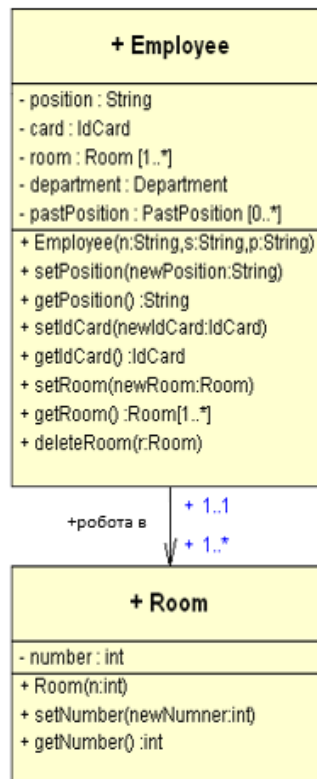


Рис. 11.11 Додавання нового класу Room

У модель додали клас «IdCard», що представляє ідентифікаційну картку (пропуск) співробітника (рис. 11.10). Кожному співробітнику може відповідати тільки одна ідентифікаційна картка, потужність зв'язку від 1 до 1.

Уявімо, що в організації належить закріплювати за працівниками приміщення. Додаємо новий клас Room (рис. 11.11). Кожному об'єкту працівник (Employee) може відповідати декілька робочих приміщень. Потужність зв'язку один-до-багатьох.

## 11.9. Відношення агрегації

Відношення агрегації має місце між декількома класами в тому випадку, якщо один з класів є деякою сутністю, що включає в себе в якості складових частин інші сутності. Дане відношення має фундаментальне значення для опису структури складних систем, оскільки застосовується для відображення системних взаємозв'язків типу «частина-ціле».

Як приклад відносини агрегації розглянемо взаємозв'язок типу «частина-ціле», яка має місце між сутністю «Вантажний автомобіль» і такими компонентами, як «Двигун», «шасі», «Кабіна», «Кузов».

Графічно відношення агрегації зображується суцільною лінією, один з кінців якої являє собою зафарбований всередині ромб (рис. 11.12). Цей ромб вказує на той з класів, який являє собою «ціле». Решта класи є його «частинами».

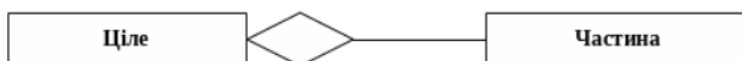


Рис. 11.12 Відношення агрегації

Прикладом відносини агрегації може служити розподіл персонального комп'ютера на складові частини: системний блок, монітор, клавіатуру і мишу.

Використовуючи позначення мови UML, компонентний склад ПК можна представити у вигляді відповідної діаграми класів, яка в даному випадку ілюструє Відношення агрегації (рис. 11.13).



Рис. 11.13 Відношення агрегації складу ПК

Введемо в модель клас Department (відділ) (рис. 11.14) - наше підприємство структуровано по відділах. У кожному відділі може працювати один або більше осіб. Можна сказати, що відділ включає в себе одного або більше співробітників і таким чином їх агрегує.

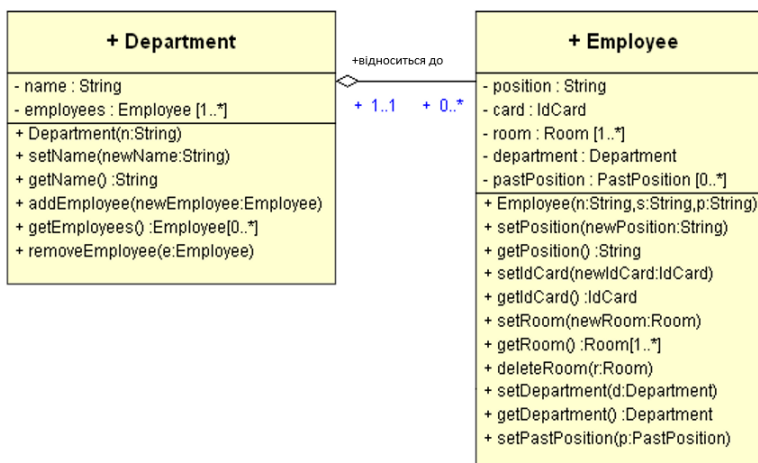


Рис. 11.14 Введення нового класу Department (відділ)

## 11.10. Відношення Композиції

Відношення композиції (рис. 11.15) є окремим випадком відносини агрегації.

Це відношення служить для виділення спеціальної форми відносини «частина-ціле», при якій складові частини в деякому сенсі перебувають всередині цілого.

Специфіка взаємозв'язку між ними полягає в тому, що частини не можуть виступати у відриві від цілого, тобто, зі знищенням цілого знищуються і всі його складові частини.



Рис. 11.15 Відношення композиції

Як додаткові позначення для відношень композиції і агрегації можуть використовуватися додаткові позначення, які використовуються для відношення асоціації. А саме, вказівка кратності класу асоціації (рис. 11.16) й імені даної асоціації, які не є обов'язковими. Стосовно описаного вище прикладу класу "Вікно\_програми" його діаграма класів може мати такий вигляд.

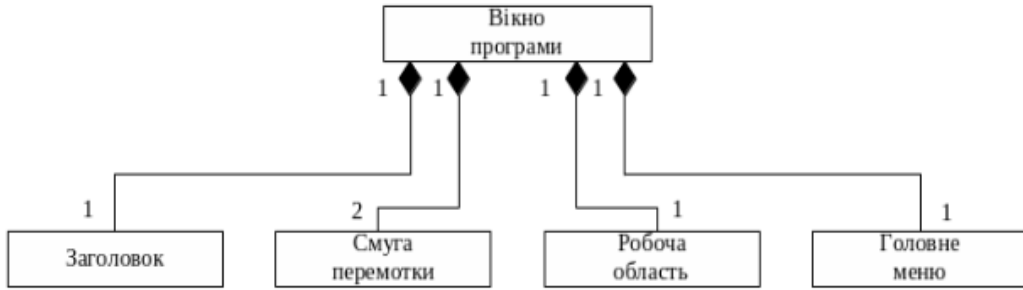


Рис. 11.16 Додаткові вказівки (кратності класу) у відношеннях композиції

Розглянемо новий клас «pastPosition». У нього, крім властивості «ім'я» (name), введемо і властивість «department», яке зв'яже його з класом «Department».

Дані про минулі займані посади є частиною даних про співробітника, таким чином між ними зв'язок ціле-частина і в водночас, дані про минулі посади не можуть існувати без об'єкту типу «Employee». Знищення об'єкта «Employee» має привести до знищення об'єктів «pastPosition» (рис. 11.17).

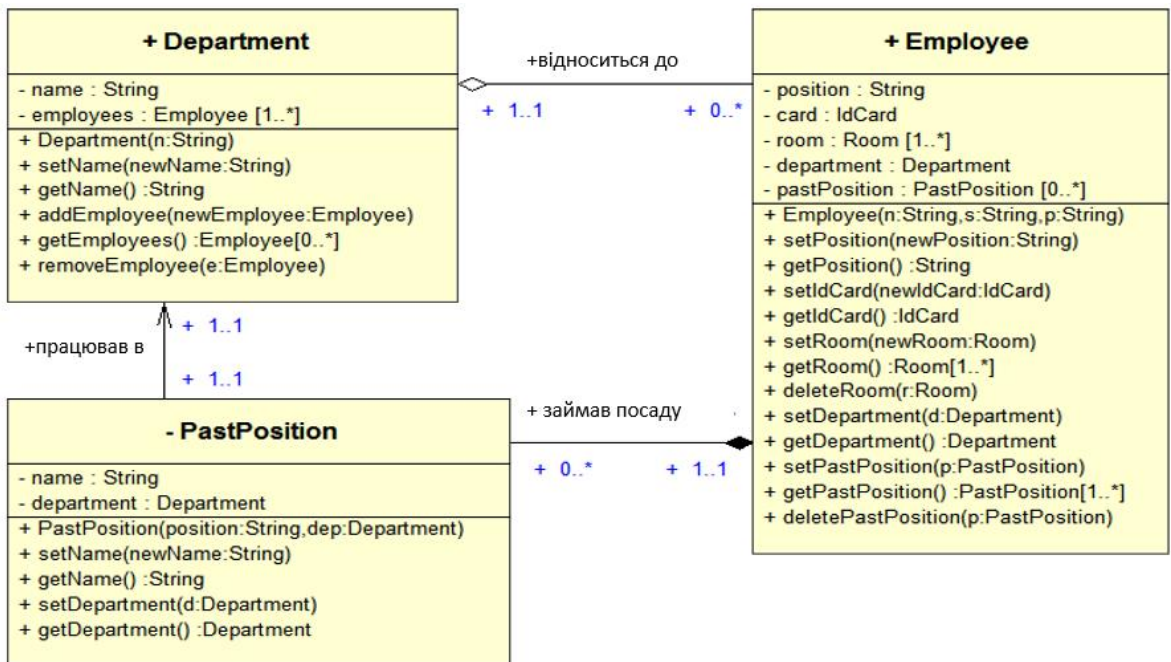


Рис. 11.17 Введення нового класу «pastPosition»

## 11.11. Відношення Узагальнення

Відношення узагальнення (рис. 11.18) є звичайним відношенням між більш загальним елементом (батьком або предком) і більш приватним або спеціальним елементом (дочірнім або нащадком).

Дане відношення може використовуватися для подання взаємозв'язків між пакетами, класами, варіантами використання і іншими елементами мови UML.

Дане відношення описує ієрархічну будову класів та успадкування їх властивостей і поведінки. При цьому передбачається, що клас-нащадок має всі властивості і поведінку класу-предку, а також має свої власні властивості і поведінку, яка відсутня у класу-предку.



Рис. 11.18 Відношення узагальнення

У цьому випадку більш загальний клас розбивається на підкласи одним відношенням Узагальнення (рис. 11.20).

Наприклад, клас Геометрична Фігура на площині (курсив позначає абстрактний клас) може виступати в якості суперкласу для підкласів, які відповідають конкретним геометричним фігурам, таким як, Прямокутник, Окружність, Еліпс і ін. Даний факт може бути представлений графічно у формі діаграми класів наступного виду (рис. 11.19).

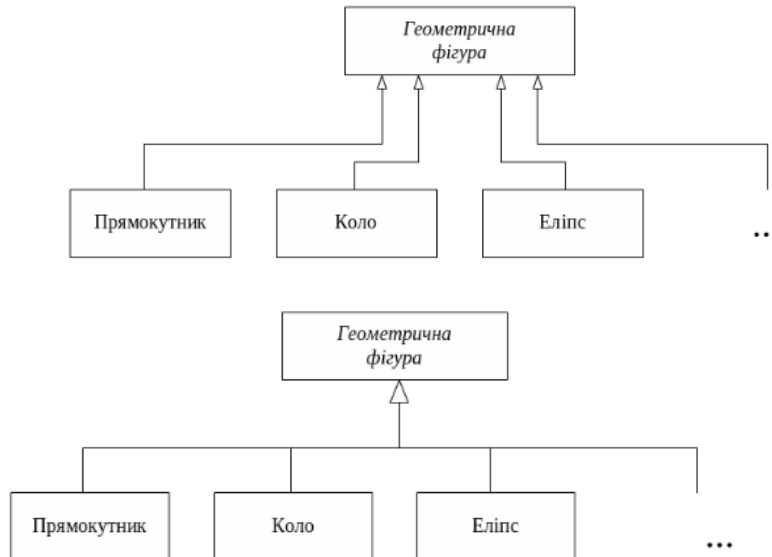


Рис. 11.19 Відношення узагальнення геометричної фігури

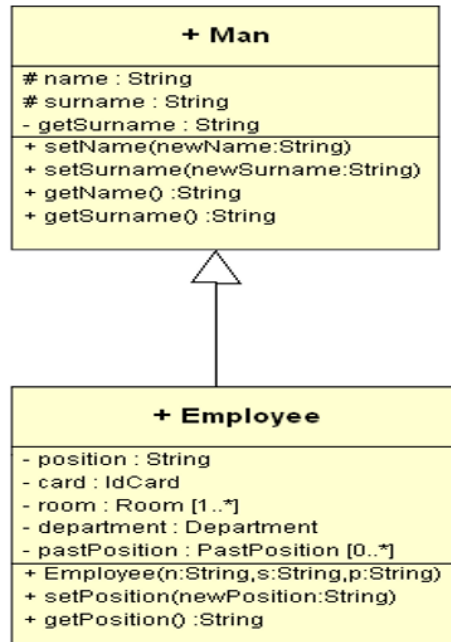


Рис. 11.20 Відношення узагальнення – це спадкування

## 11.12. Відношення Реалізація

Реалізація (realization) (рис. 11.21) - це семантичне відношення між класифікаторами, при якому один класифікатор визначає зобов'язання, а інший гарантує його виконання.

Відношення реалізації зустрічаються в двох випадках:

1. по-перше - між інтерфейсами які реалізують їх класами або компонентами;
2. а по-друге - між прецедентами які реалізують їх кооперації (конкретними зв'язками між конкретними класами - так наприклад "співпрацювати" можуть як дві компанії так і компанія і конкретна людина).

Відношення реалізації зображується у вигляді пунктирною лінії з незафарбовані стрілкою (трикутником на кінці), як щось середнє між відносинами узагальнення і залежності - стрілка спрямована на "шаблон" - тобто на ту сутність, яка задає правила власної реалізації для сутності (реалізована), яка ці правила виконує (реалізує).

Для демонстрації відносини «реалізація» створимо інтерфейс «Unit». Якщо уявити, що організація може ділитися не тільки на відділи, а наприклад, на цеху, філії і т.ін. Інтерфейс «Unit» являє собою саму абстрактну одиницю поділу. У кожній одиниці поділу працює якась кількість співробітників, тому метод для отримання

кількості працюючих людей буде актуальний для кожного класу, що реалізує інтерфейс «Unit».

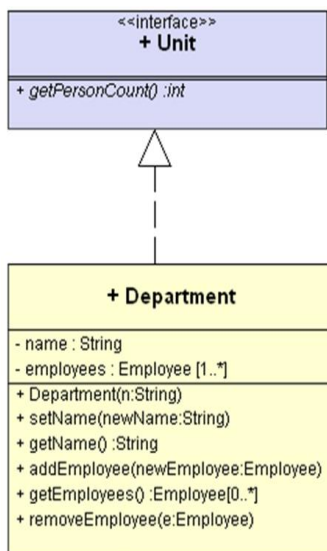


Рис. 11.21 Відношення реалізація

### 11.13. Інтерфейси

Інтерфейс (Interface) – це сукупність операцій, які визначають сервіс (набір послуг), що надається класом або компонентом. Таким чином, інтерфейс описує видиму ззовні поведінку елемента.

Інтерфейс може представляти поведінку класу або компонента повністю або частково; він визначає тільки специфікації операцій (сигнатури), але ніколи - їх реалізації.

Однак при побудові діаграми класів окремі інтерфейси можуть уточнюватися (рис. 11.22) і в цьому випадку для їх зображення використовується спеціальний графічний символ - прямокутник класу з ключовим словом або стереотипом «interface».

При цьому секція атрибутів у прямокутника відсутня, а вказується лише секція операцій.

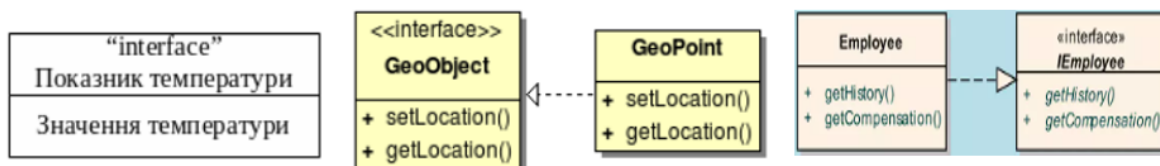


Рис. 11.22 Уточнення Інтерфейсу в діаграмі класів

## 11.14. Об'єкти

Об'єкт (object) є окремим екземпляром класу, який створюється на етапі виконання програми.

Він має своє власне ім'я і конкретні значення атрибутів.

В силу різних причин може виникнути необхідність показати взаємозв'язок не тільки між класами моделі, а й між окремими об'єктами, що реалізують ці класи.

Для графічного зображення об'єктів використовується такий же символ прямокутника, що і для класів.

Відмінності проявляються при вказівці імен об'єктів, які в разі об'єктів обов'язково підкреслюються.

При цьому запис імені об'єкта є рядком тексту «ім'я об'єкта: ім'я класу», розділену двокрапкою (рис. 11.23.а, в).

Ім'я об'єкта може бути відсутнім, в цьому випадку передбачається, що об'єкт є анонімним, і двокрапка вказує на дану обставину (рис. 11.23.г).

Відсутнім може бути і ім'я класу. Тоді вказується просто ім'я об'єкта (рис. 11.23.б). Атрибути об'єктів приймають конкретні значення (рис. 11.24).

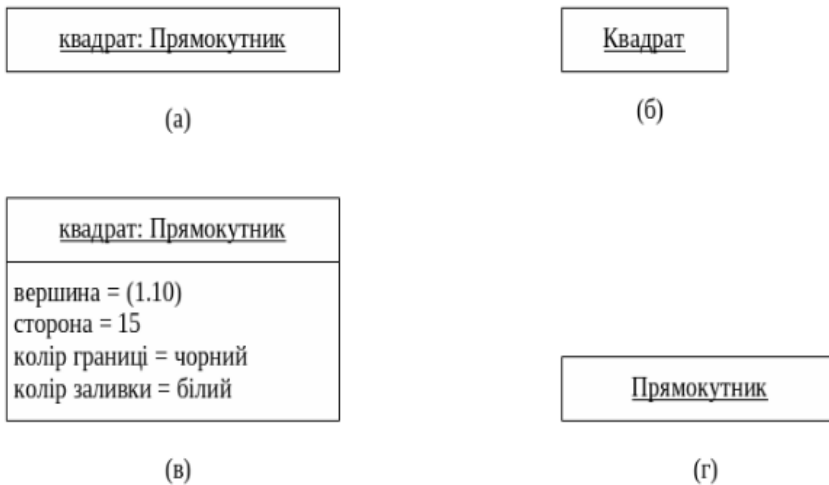


Рис. 11.23 Об'єкти: а, в - «ім'я об'єкта: ім'я класу»; б - «ім'я класу»; г - «ім'я об'єкта»

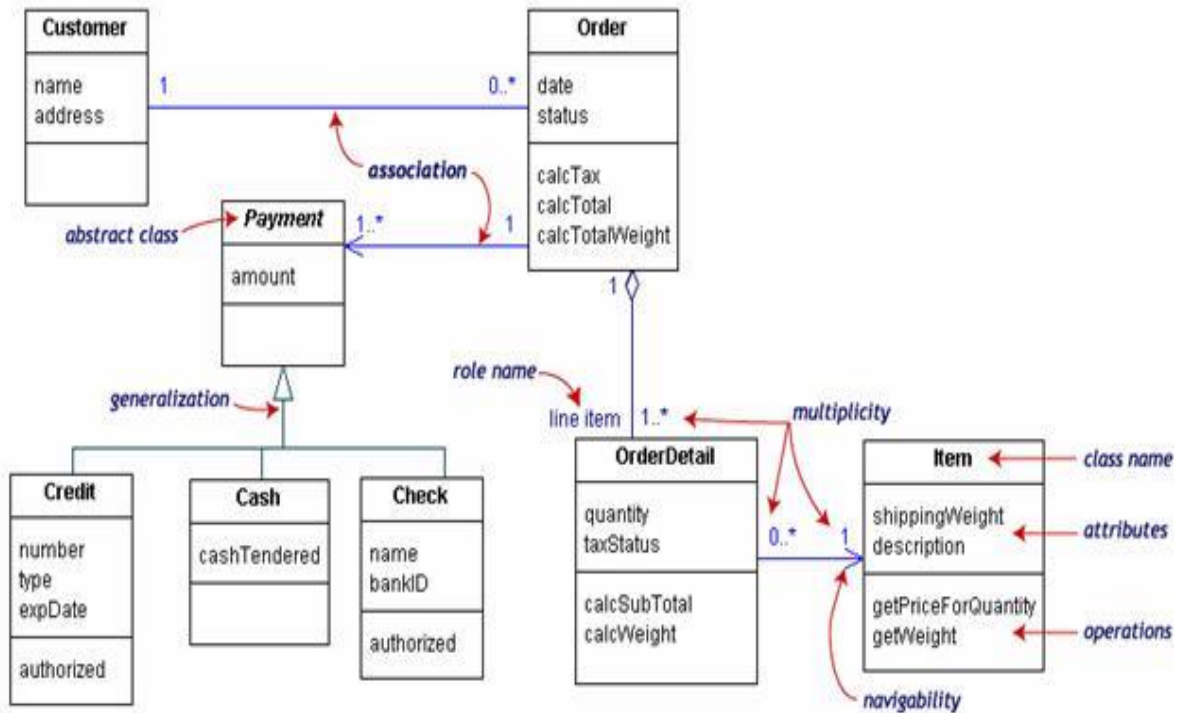


Рис. 11.24 Приклад діаграми класів

## Питання до розділу 11

1. Для чого служить діаграма класів (Class Diagram)?
2. Яким чином описується клас у Class Diagram?
3. Яким чином описується об'єкти та інтерфейси у Class Diagram?
4. Як описуються атрибути класу у Class Diagram?
5. Як описуються операції класу у Class Diagram?
6. Які є типи відношень між класами у UML?

## РОЗДІЛ 12. ПОСЛІДОВНІСТЬ ВИКОНАННЯ ДІЙ БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ

### 12.1. Діаграма станів (statechart diagram)

Діаграма класів представляє собою логічну модель статичного уявлення модельованої системи [22]. На даній діаграмі зображуються тільки взаємозв'язки структурного характеру, які не залежать від часу або реакції системи на зовнішні події.

Однак для більшості фізичних систем, крім найпростіших і тривіальних, статичних уявлень абсолютно недостатньо для моделювання процесів функціонування подібних систем як в цілому, так і їх окремих підсистем і елементів.

Будь-яка складна програма, програмно-технічний комплекс, технічний пристрій, такий як комп'ютер, смартфон, планшет, автомобіль в найзагальнішому випадку може характеризуватися такими своїми станами, як “справний” і “несправний”, “у робочому стані” чи “у стані наладки” та таке інше.

Для моделювання поведінки на логічному рівні в мові UML можуть використовуватися відразу кілька діаграм:

1. станів;
2. діяльності;
3. послідовності;
4. кооперації.

На відміну від інших діаграм, діаграма станів описує процес зміни станів тільки одного класу, а точніше – одного екземпляра певного класу, тобто моделює всі можливі зміни в стані конкретного об'єкта.

Діаграма станів по суті є графом спеціального виду, який представляє певний кінцевий автомат.

Поняття автомата в контексті UML має досить специфічну семантику, засновану на теорії автоматів.

Вершинами цього графа являються стани і деякі інші типи елементів автомата (псевдостани), які зображуються відповідними графічними символами. Дуги графа слугують для позначення переходів зі стану в стан.

## 12.2. Автомати

Автомат (state machine) в мові UML являє собою певний формальний опис (формалізм) для моделювання поведінки елементів моделі і системи в цілому.

В метамоделі UML автомат є пакетом, в якому визначено безліч понять, необхідних для опису поведінки модельованої сутності у вигляді дискретного простору з кінцевим числом станів і переходів.

З іншого боку, автомат описує поведінку окремого об'єкта в формі послідовності станів, які охоплюють всі етапи його життєвого циклу, починаючи від створення об'єкта і закінчуючи його знищенням.

Кожна діаграма станів представляє певний автомат – структуру, що описує можливі шляхи зміни поточних станів об'єктів та систем (технічних, програмних, програмно-технічних систем, людино-машинних систем та інші).

Найпростішим прикладом візуального представлення станів і переходів на основі формалізму автоматів може служити ситуація з справністю технічного пристрою, такого як комп'ютер.

В цьому випадку вводяться в розгляд два найзагальніших стани: «справний» і «несправний» і два переходи: «вихід з ладу» і «ремонт» (рис 12.1).

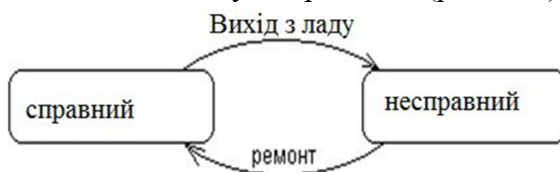


Рис. 12.1 Два стани автомату

Передбачається для простоти опису, що час переходу з одного стану в інший дорівнює нулю (якщо додатково нічого не сказано).

Іншими словами, перехід об'єкта зі стану в стан відбувається миттєво.

Формалізм звичайного автомата заснований на виконанні наступних обов'язкових умов:

1. Автомат не запам'ятовує історію переміщення із стану в стан. З точки зору моделювання поведінки визначальним є сам факт знаходження об'єкта в тому чи іншому стані, але ніяк не послідовність станів, в результаті якої об'єкт перейшов в поточний стан.

Іншими словами, автомат «забуває» всі попередні стани, які передували поточному в даний момент часу.

2. У кожен момент часу автомат може перебувати в одному і тільки в одному зі своїх станів. Це означає, що формалізм автомата призначений для моделювання

послідовної поведінки, коли об'єкт протягом свого життєвого циклу послідовно проходить через усі свої статки.

При цьому автомат може перебувати в окремому стані як завгодно довго, якщо не відбувається ніяких подій (стабільність стану).

3. Хоча процес зміни станів автомата відбувається в часі, явно концепція часу не входить в формалізм автомата. Це означає, що тривалість перебування автомата в тому чи іншому стані, а також час досягнення того чи іншого положення не специфікується.

Час на діаграмі станів присутній в неявному вигляді, хоча для окремих подій може бути вказаний інтервал часу і в явному вигляді.

4. Кількість станів автомата повинно бути обов'язково кінцевим (в мові UML розглядаються тільки кінцеві автомати), і всі вони повинні бути специфіковані явно.

При цьому окремі псевдостани можуть не мати специфікацій (початковий і кінцевий стани). В цьому випадку їх призначення і семантика повністю визначаються з контексту моделі і розглянутої діаграми станів.

### 12.3. Стан

Поняття стану (state) є фундаментальним не тільки в метамоделі мови UML, але і в прикладному системному аналізі.

Стан може бути задано у вигляді набору конкретних значень атрибутів класу або об'єкта (рис. 12.2), при цьому зміна їх окремих значень буде відображати зміна стану модельованого класу або об'єкта.

Слід зауважити, що не кожен атрибут класу може характеризувати його стан.

Як правило, мають значення тільки такі властивості елементів системи, які відображають динамічний або функціональний аспект її поведінки.



Рис. 12.2 Стани автомату: а – ім'я стану; б – ім'я стану разом з його списком дій або переходів

Стан на діаграмі зображується прямокутником із закругленими вершинами. Цей прямокутник, в свою чергу, може бути розділений на дві секції горизонтальною лінією.

Якщо вказана лише одна секція, то в ній записується тільки ім'я стану (рис. 12.2 а).

В іншому випадку в першій з них записується ім'я стану, а в другій - список деяких внутрішніх дій або переходів в даному стані.

При цьому під дією в мові UML розуміють деяку операцію, виконання якої призводить до зміни стану або поверненню деякого значення (наприклад, «правда» або «неправда»).

Ім'я стану являє собою рядок тексту, яка розкриває змістовний сенс даного стану.

Ім'я завжди записується з великої літери.

Список внутрішніх дій містить перелік можливих внутрішніх дій або діяльностей, які виконуються в процесі знаходження модельованого елемента в даному стані.

Кожна з дій записується у вигляді окремого рядка і має такий вигляд:

<Мітка-дії '/' вираз-дії>

Мітка дії вказує на обставини або умови, при яких буде виконуватися діяльність, визначена виразом дії.

Перелік міток дії має фіксовані значення в мові UML, які не можуть бути використані в якості імен подій. Ці значення наступні:

entry – ця мітка вказує на дію, специфіковану наступним за нею виразом дії, яка виконується в момент входу в даний стан (вхідна дія);

exit – ця мітка вказує на дію, специфіковану наступним за нею виразом дії, яка виконується в момент виходу з цього стану (вихідна дія);

do - ця мітка описує виконання дії («do activity»), яка виконується протягом всього часу, поки об'єкт знаходиться в даному стані.

При завершенні дії генерується відповідний результат.

include - ця мітка використовується для звернення до підавтомату, при цьому наступний за нею вираз дії містить ім'я цього підавтомату (декомпозиція дій).

Як приклад стану розглянемо ситуацію введення пароля користувача при аутентифікації входу в деяку програмну систему (рис. 12.4). В цьому випадку список внутрішніх дій в даному стані не порожній і включає 4 окремих дії, перші дві з яких стандартні, а два останніх визначаються своєю специфікацією.



Термін подія (event) є самостійним елементом мови UML. Формально, подія є специфікацію деякого факту, що має місце в просторі і в часі.

Поняття події (рис. 12.5) фіксує увагу на зовнішніх проявах якісних змін, що відбуваються при переході об'єкта, що моделюється зі стану в стан.

Наприклад, при включенні електричного перемикача відбувається деяка подія, в результаті якого кімната стає освітленою. Після успішного ремонту комп'ютера також відбувається важлива подія – відновлення його працездатності. Якщо підняти трубку звичайного телефону, то, в разі його справності, ми очікуємо почути звуковий сигнал. І цей факт теж є подією. На рис. 12.6 зображено приклад подій з ADSL підключення.

У мові UML події грають роль активаторів, які ініціюють переходи з одних станів в інші.

Як події можна розглядати сигнали, виклики, закінчення фіксованих проміжків часу або моменти закінчення виконання певних дій (подійно-орієнтовані мікросервісні системи).



Рис. 12.6 Приклад з *adsl* підключенням

Сторожова умова (guard condition), якщо воно є, завжди записується в дужках після події-тригера і являє собою деякий булевий вираз, що має приймати одне з двох взаємно виключають значень: «так» або «ні».

Обчислення істинності сторожової умови відбувається тільки після виникнення асоційованого з ним події-тригера, що ініціює відповідний перехід.

Складений стан (composite state) (рис. 12.7) - складний стан, який складається з інших вкладених в нього станів. Останні будуть виступати по відношенню до першого як підстані (substate).

Хоча між ними має місце відношення композиції, графічно всі вершини діаграми, які відповідають вкладеним станам, зображуються усередині символу складеного стану.



Рис. 12.7 Складений стан

Послідовні підстани (sequential substates) (рис. 12.8) використовуються для моделювання такої поведінки об'єкта, під час якого в кожен момент часу об'єкт може знаходитися в одному і тільки одному з підстанів. Поведінка об'єкта в цьому випадку представляє собою послідовну зміну підстанів, починаючи від початкового і закінчуючи кінцевим підстанів.

Для прикладу розглянемо в якості об'єкта, що моделюється звичайний телефонний апарат. Він може знаходитися в різних станах, одним з яких є стан дозвону до абонента. Очевидно, для того щоб зателефонувати, необхідно зняти телефонну трубку, почути звуковий сигнал, після чого набрати потрібний телефонний номер. Таким чином, стан дозвону до абонента є складовим і складається з двох послідовних підстанів: «підняти телефонну трубку» і «набрати телефонний номер».

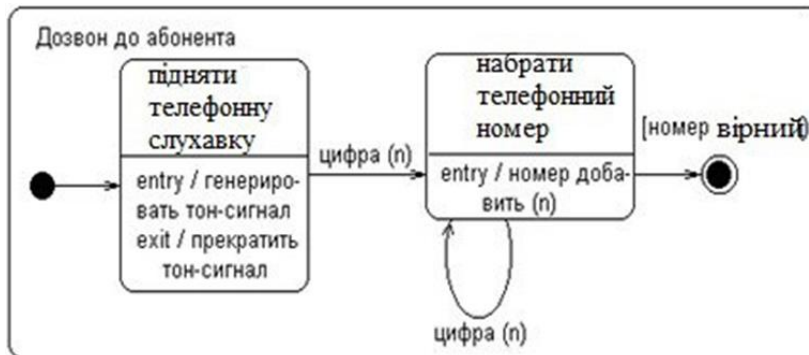


Рис. 12.8 Підстани – дозвон до абонента

Паралельні підстани (concurrent substates) дозволяють специфікувати два і більше підавтоматів, які можуть виконуватися паралельно всередині складеного події. Кожен з підавтоматів займає деяку область (регіон) всередині складеного стану, яка відділяється від решти горизонтальною пунктирною лінією.

Якщо на діаграмі станів є складений стан з вкладеними паралельними підстанами, то об'єкт може одночасно перебувати в кожному з цих підстанів (рис. 12.9).

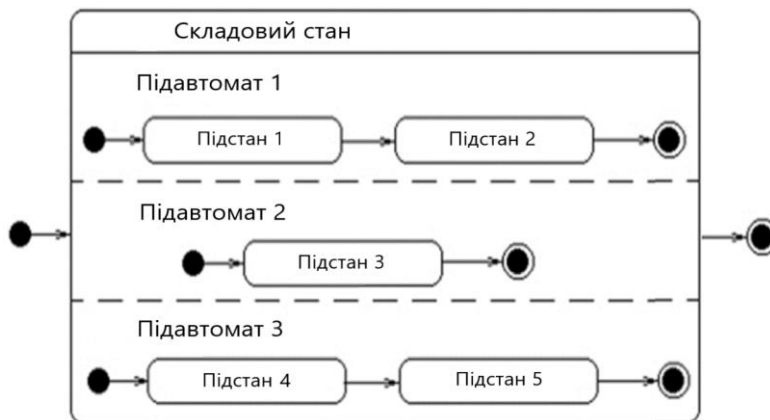


Рис. 12.9 Паралельні підстанни

В окремих випадках перехід може мати кілька станів-джерел і кілька цільових станів (рис. 12.10). Такий перехід отримав спеціальну назву – паралельний перехід.

Введення в розгляд паралельного переходу обумовлено необхідністю синхронізувати і / або розділити окремі підпроцеси на паралельні напрямки без специфікації додаткової синхронізації паралельних підавтоматів.

Приклад: автомобіль не заведений і світло фар не горить, після заводу двигуна спалахують фари ...

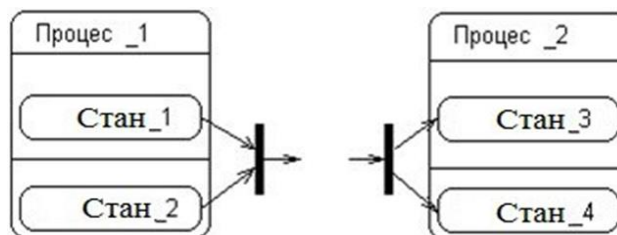


Рис. 12.10 Декілька станів-джерел переходу

Приклад. Розглянемо спрощену ситуацію з моделюванням процесу спорудження будинку. Припустимо, що побудова будинку включає в себе будівельні роботи (зведення фундаменту і стін, зведення даху та оздоблювальні роботи) і роботи з електрифікації будинку (підведення електричної лінії, прокладка прихованої електропроводки і установка освітлювальних ламп). Очевидно, два цих комплексу робіт можуть виконуватися паралельно, однак між ними є деякий взаємозв'язок.

Зокрема, прокладка прихованої електропроводки може початися лише після того, як буде завершено зведення фундаменту і стін. А оздоблювальні роботи слід почати лише після того, як буде закінчена прокладка прихованої електропроводки. В іншому випадку оздоблювальні роботи доведеться проводити повторно.

На рис. 12.11 зображено приклад обробки транзакцій оплати.

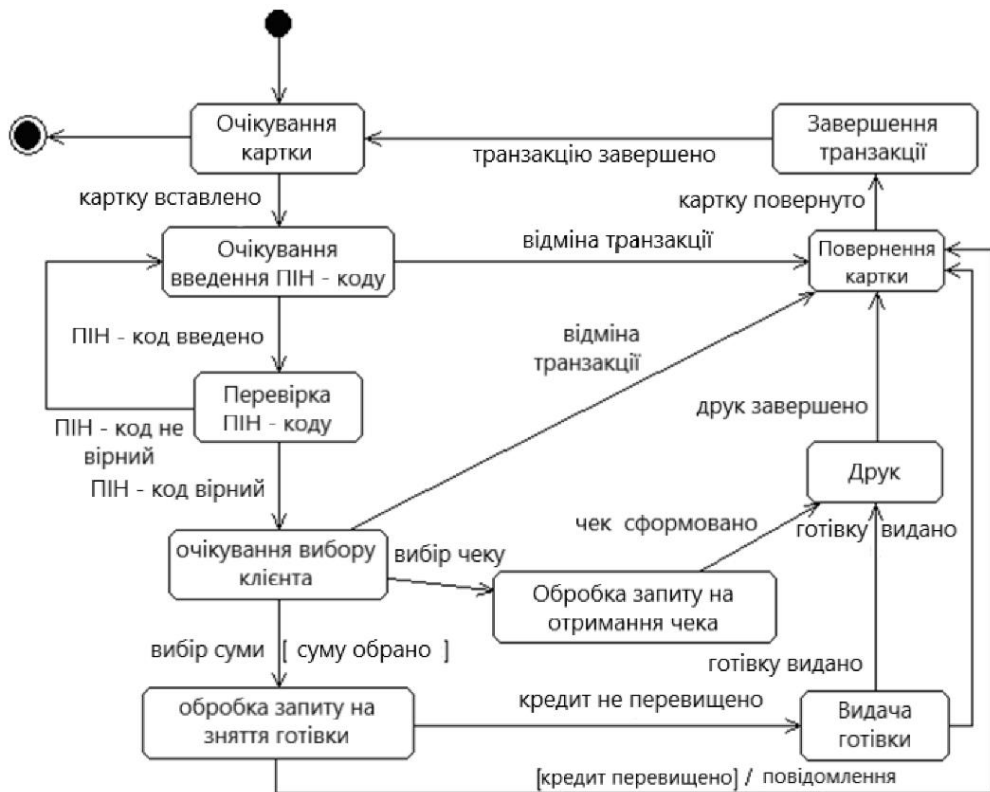


Рис. 12.11 Приклад. Обробка транзакцій

## Питання до розділу 12

1. Що представляє собою діаграма станів (Statechart Diagram)?
2. В чому суть формалізму автомату у діаграмі станів?
3. Які обов'язкові умови формалізму автомата у діаграмі станів?
4. Як описується стан у діаграмі станів?
5. Які основні символи використовуються у діаграмі станів?
6. Яким чином описується синхронізація процесів у діаграмі станів?

## **РОЗДІЛ 13. ПРОЦЕСНИЙ АНАЛІЗ У ПОБУДОВІ СЦЕНАРІЇВ**

### **13.1. Процесний підхід**

Головне поняття, яке використовує процесний підхід – це поняття процесу. Відповідно до концепції процесного підходу вся діяльність організацій розглядається як набір процесів [1,18].

Існують різні визначення, але найчастіше використовується визначення стандарту ISO 9001.

Процес – це сукупність взаємопов'язаних та взаємодіючих видів діяльності, які перетворюють входи у виходи.

Важливою складовою процесу, яка не відображена в цьому визначенні, є систематичність дій.

Дії процесу мають бути повторюваними у певній послідовності, а не випадковими.

### **13.2. Цілі процесного підходу**

Як правило, на етапі бізнес-моделювання для рішення задач інформатизації організацій ставиться завдання досягнення низки цілей, серед яких можна виділити:

- підвищення якості обслуговування;
- скорочення загальної тривалості циклу процесу;
- підвищення продуктивності;
- скорочення часу очікування;
- зниження витрат на здійснення даної діяльності;
- ефективне використання трудових ресурсів тощо.

Запропоновані цілі, як правило, повинні співпадати з основними цілями розвитку компанії і тому логічним першим ланцюгом моделювання є побудова дерева цілей розвитку підприємства, яке і буде основою для виділення та ранжування цілей для моделювання бізнес-процесів та сценаріїв окремих процесів.

Це потребує перехід від AS IS до TO BE, що передбачає реінжиніринг бізнес-процесів об'єкту інформатизації.

### **13.3. Процесне управління**

Процесне управління — це сучасна концепція менеджменту, яка розглядає діяльність будь-якого підприємства як сукупність бізнес-процесів та надає програмні інструменти для оптимізації діяльності підприємства.

Інакше такий підхід називається BPM - Business Process Management.

Згідно вимог Міжнародного стандарту ISO 9001 (“Системи менеджменту якості. Вимоги”) основою у побудові і функціонуванні систем управління організаціями і підприємствами є методика процесного підходу. (Сертифікація ISO 9001- це важливо).

Управління підприємством, чи організацією із врахуванням вимог вищезгаданого стандарту, дозволяє керівництву побудувати систему ефективного керування бізнесом і організацією в цілому та кожним її процесом зокрема, інтегрувати окремі процеси в єдину систему, адаптувати у її функціонування механізм безперервних вдосконалення.

Процесний підхід є відходом від раніше домінуючого, підходу, що традиційно був заснований на ієрархічно-функціональних схемах управління.

### **13.4. Мета процесного підходу**

Процесний підхід був розроблений і застосовується з метою створення горизонтальних зв'язків в підрозділах організацій. Підрозділи та співробітники, задіяні в одному процесі, можуть самостійно координувати роботу в рамках процесу та вирішувати проблеми, що виникають без участі вищого керівництва. Процесний підхід до управління дозволяє більш оперативно вирішувати питання, що виникають, і впливати на результат.

На відміну від функціонального підходу, управління процесами дозволяє концентруватися не так на роботі кожного з підрозділів, а на результатах роботи організації загалом.

Процесний підхід змінює поняття структури бізнес-процесів та вводить поняття процесів та їх сценаріїв, що є в організації.

Основним елементом стає процес.

Відповідно до одного з принципів процесного підходу організація складається не з підрозділів, а з процесів.

### 13.5. Модель управління підприємством

На рис. 13.1 зображено узагальнену процесну модель підприємства, а на рис 13.2 загальну модель управління підприємством.

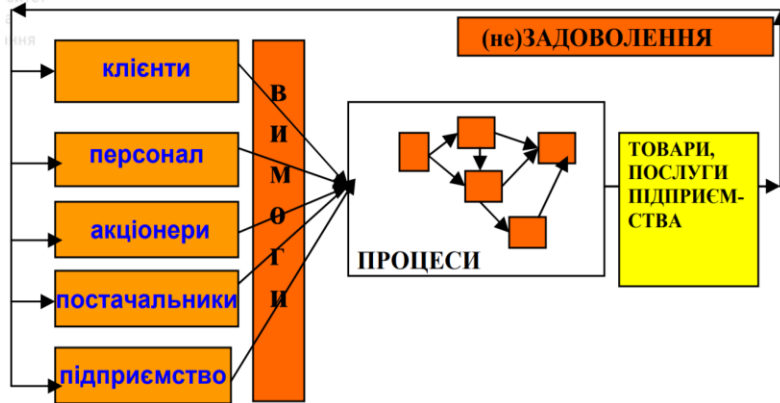


Рис. 13.1 Загальна процесна модель управління підприємством



Рис. 13.2 Загальна модель управління підприємства

### 13.6. Переваги процесного підходу

За рахунок того, що процесний підхід створює горизонтальні зв'язки в роботі організації, він дозволяє отримати низку переваг у порівнянні з функціональним підходом.

Основними перевагами процесного підходу є:

- координація дій різних підрозділів у рамках процесу;
- орієнтація на результат процесу;
- підвищення результативності та ефективності роботи організації;
- прозорість дій щодо досягнення результату;
- підвищення передбачуваності результатів;
- виявлення можливостей для цілеспрямованого покращення процесів;
- усунення бар'єрів між функціональними підрозділами;
- скорочення зайвих вертикальних взаємодій;
- виключення незатребуваних процесів;
- скорочення тимчасових та матеріальних витрат.

### 13.7. Перехід до процесного підходу

Перехід до процесного підходу в управлінні складається з 3-х великих кроків:

1. Регламентация бізнес-процесів, яка включає визначення спершу стратегії бізнесу (якщо її немає), а потім перехід до бізнес-процесів і їх вирівнювання / дизайн під стратегію, що розроблено. Як наслідок — зміна організаційної структури, перевизначення профілів необхідних фахівців під новий підхід до управління.
2. Інформатизація, створення керованої ІТ моделі бізнес-процесів і постійне підвищення їх ефективності.
3. Побудова необхідних та можливих сценаріїв дій (протоколів) для кожного з процесів.

### 13.8. Зв'язок процесів та сценаріїв

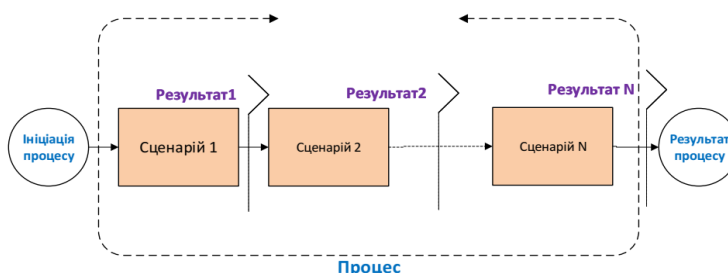


Рис. 13.3 Зв'язок процесів та сценаріїв

Найчастіше сценарії в бізнес-процесах аналізують з використанням графу (рис. 13.4).

Сценарії (протоколи), як правило, описують однонаправлені (без циклів та зворотних зв'язків) послідовності кроків з визначенням конкретних виконавців (ролей) (рис. 13.5).

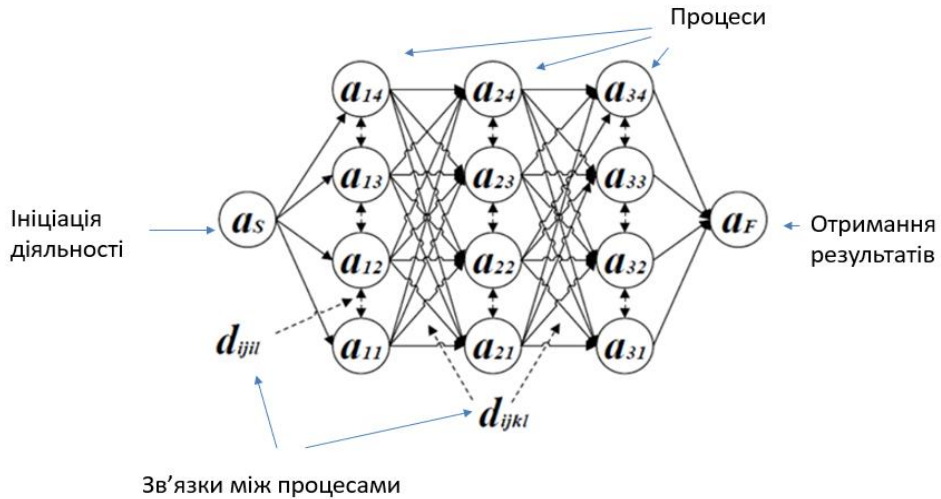


Рис. 13.4 Процесна модель підприємства на основі графу

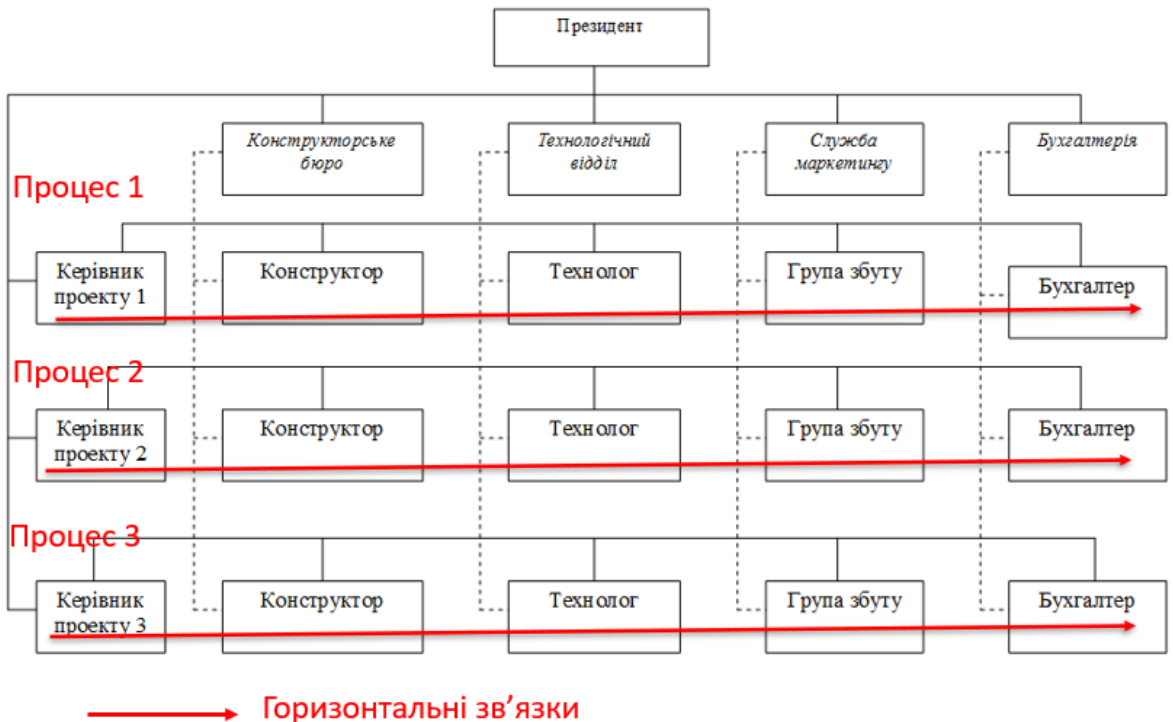


Рис. 13.5 Участь виконавців у процесах в організації з ієрархічною структурою

### **13.9. Застосування процесного підходу у побудові та управлінні організацією**

При застосуванні процесного підходу, як методу забезпечення життєдіяльності системи управління, надається можливість передбачати:

- 1) розуміння і виконання вимог споживача;
- 2) необхідність розгляду процесів з точки зору споживчої цінності, створеної у продукті дії;
- 3) досягнення результативності і ефективності у робочих характеристиках процесів;
- 4) постійне покращення процесів та сценаріїв виконання окремих дій (протоколів), що базується на об'єктивних даних отриманих внаслідок вимірювання стану їх протікання.

### **13.10. Ключові елементи процесного підходу**

Входами процесу є елементи, які зазнають змін під час виконання дій. Як входи процесний підхід розглядає матеріали, обладнання, документацію, різну інформацію, персонал, фінанси та ін.

Виходами процесу є очікувані результати, заради яких виконуються дії. Виходом може бути як матеріальний продукт, так і різноманітні послуги або інформація.

Ресурсами є елементи, необхідні процесу. На відміну від входів, ресурси не змінюються у процесі. Такими ресурсами процесний підхід визначає обладнання, документацію, фінанси, персонал, інфраструктуру, середовище та ін.

Власник процесу - процесний підхід вводить це поняття як одне з найголовніших. У кожного процесу має бути свій власник. Власником є людина, яка має у своєму розпорядженні необхідну кількість ресурсів і відповідає за кінцевий результат (вихід) процесу. Відповідає за виконання сценаріїв дій окремих виконавців.

Постачальники та споживачі є у кожного процесу. Постачальники забезпечують вхідні елементи процесу, а споживачі зацікавлені в отриманні вихідних елементів. У процесу можуть бути як зовнішні, так і внутрішні постачальники та споживачі. Якщо у процесу немає постачальників, процес не буде виконано. Якщо процес немає споживачів, то процес не є необхідним.

Показники процесу необхідні для отримання інформації про його роботу та прийняття відповідних управлінських рішень. Показники процесу це набір кількісних чи якісних параметрів, що характеризують процес і його результат (вихід).

### 13.11. Вдосконалення діяльності на основі процесного підходу

Чотири напрями, що використовують процесний підхід як головний підхід щодо підвищення ефективності діяльності:

1. Загальний менеджмент якості (TQM). Це концепція, яка передбачає безперервне підвищення якості продукції, процесів та системи управління організацією. У основу роботи організації ставиться задоволення споживача;

2. Постійне покращення процесів (Continuous Improvement Process). Це концепція, яка передбачає незначні, але постійні поліпшення процесу за всіма його складовими. Найбільш відомим підходом, в основі якого лежить постійне покращення процесів є японський підхід Кайдзен (kaizen) (рис. 13.6);



Рис. 13.6 Японський підхід Кайдзен (kaizen). Цикл постійного покращення процесів

3. Удосконалення бізнес-процесів (Business Process Improvement) або управління бізнес-процесами (Business Process Management). Це підхід, спрямований на те, щоб допомогти організаціям оптимізувати бізнес-процеси з метою підвищення їх ефективності. Зміни процесів здійснюються поступово, але обов'язково на систематичній основі;

4. Реінжиніринг бізнес-процесів (Business Process Reengineering). У його основі лежить переосмислення існуючих процесів та його радикальне зміна (перепроєктування). На відміну від трьох вищезгаданих підходів реінжиніринг передбачає швидку зміну процесів та сценаріїв виконання дій (протоколів). Також у цьому підході значний акцент робиться на застосування інформаційних технологій.

### 13.12. Висновок

Отже, на сучасному етапі інформатизації та розвитку підприємств та організацій, застосування процесного підходу на рівні будь-якого підприємства дає можливість ефективного управління на основі розробки та впровадження цілісної системи процесів та сценаріїв діяльності.

### **Питання до розділу 13**

1. Яка мета та цілі процесного підходу?
2. Що таке процесне управління?
3. В чому сутність моделі управління підприємством?
4. Які переваги процесного підходу?
5. В чому суть зв'язку процесів та сценаріїв?
6. Як будується процесна модель на основі графу?
7. Які ключові елементи процесного підходу?

## РОЗДІЛ 14. АНАЛІЗ БІЗНЕС-ДІЯЛЬНОСТІ

### 14.1. ERP

ERP-система (Enterprise Resource Planning System – рис. 14.1, 14.2, 14.3) – система планування ресурсів підприємства – це корпоративна інформаційна система (КІС), призначена для автоматизації управління організацією, підприємством чи групою підприємств.

Більшість ERP - систем побудовані по модульному принципу, і в тій чи іншій ступені охоплюють всі ключові процеси діяльності компанії [2, 3, 5].

ERP – це системи управління ключовими бізнес-процесами підприємства.

ERP-система включає в себе модулі:

- планування діяльності компанії;
- бюджетування та контролінгу;
- логістика;
- ведення обліку;
- управління персоналом;
- управління виробництвом;
- управління клієнтами.



Рис. 14.1 ERP-система

ERP система – це комплексне сховище всіх видів інформації про діяльність організації, що дає можливість отримання даних за напрямками діяльності організації в рамках роботи в одній системі усіх функціонально-організаційних підрозділів.

## ERP-система у комплексі систем



Рис. 14.2 ERP-система у комплексі систем

### 14.2. ERP-system

Класична технологія опису бізнес-діяльності підприємств та організацій з метою побудови чи реінжинірингу інформаційної системи.

Етап 1. Опис організаційної структури.

Етап 2. Опис фінансової структури.

Етап 3. Опис інформаційної системи.

Етап 4. Опис функцій підрозділів.

Етап 5. Опис продуктів та послуг.

Етап 6. Виділення і побудова дерева бізнес-процесів.

Етап 7. Оцінка та вибір бізнес-процесів для подальшого опису, аналізу та оптимізації (реінжинірингу).

Етап 8. Опис інформаційних потоків, обраних бізнес-процесів.

Етап 9. Опис матеріальних потоків, обраних бізнес-процесів.

Етап 10. Опис обраних бізнес-процесів.

Етап 11. Збір форм документів та встановлення зв'язків з графічними відображеннями моделей бізнес-процесів та інформаційних потоків.



Рис. 14.3 ERP-система

Етап 1. "Опис організаційної структури".

В рамках етапу "Опис організаційної структури" розробляється кілька моделей організаційної структури, основними з яких є моделі, побудовані за принципом підпорядкованості та принципу входження (рис. 14.4).

При описі організаційної структури визначається перелік підрозділів та посад, існуючих в об'єкті інформатизації, а також визначається їх ієрархічна взаємопов'язаність. Цей етап необхідний для визначення функцій та кількості інформатизованих робочих місць для інформатизованої системи, що розробляється. Необхідно для оцінки архітектури ПЗ та вартості проекту інформатизації взагалі.

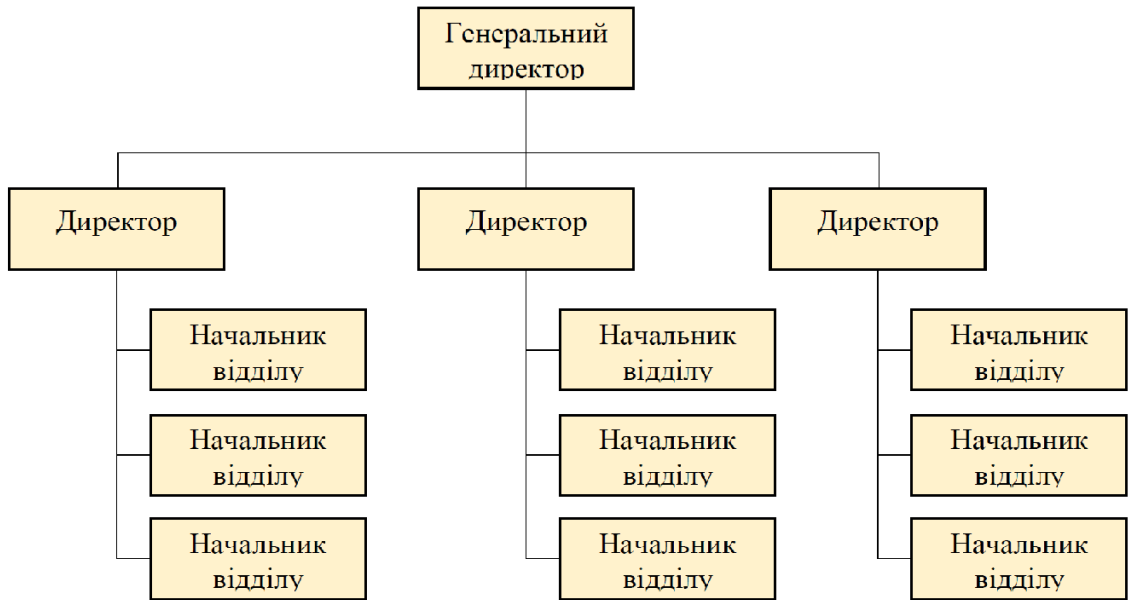


Рис. 14.4 Лінійно-функціональна організаційна структура

## Етап 2. "Опис фінансової структури".

В рамках даного етапу будується модель фінансової структури, при розробці якої ідентифікуються типи центрів фінансової відповідальності (ЦФВ) (рис. 14.5), існуючих в компанії, описується їх перелік, а також описується склад структурних підрозділів компанії, що входять в той чи інший центр фінансової відповідальності.

Цей етап необхідний для визначення фінансової безпеки підприємства. На основі цього етапу будуються сценарії роботи центрів фінансової відповідальності для подальшої їх інформатизації, щоб виключити можливість яких-будь можливих фінансових помилок та махінацій.

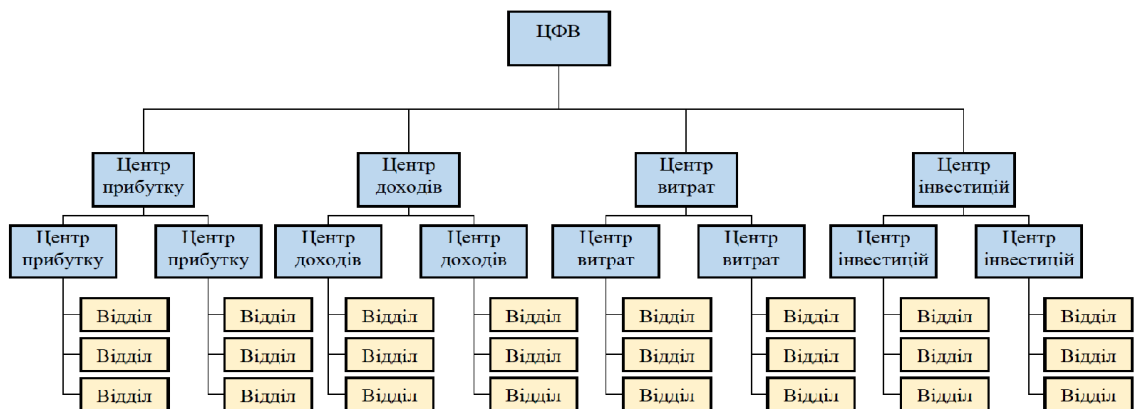


Рис. 14.5 Центр фінансової відповідальності

### Етап 3. "Опис інформаційної системи"

В рамках даного етапу будується модель інформаційної системи, при розробці якої показуються типи інформаційних систем (рис. 14.6), які застосовуються в компанії, описується їх модульна структура, а також перераховується програмне забезпечення, що використовується в компанії при виконанні бізнес-процесів.

Таким чином визначається можливість використання вже існуючого ПЗ та необхідності його модернізації чи заміни.

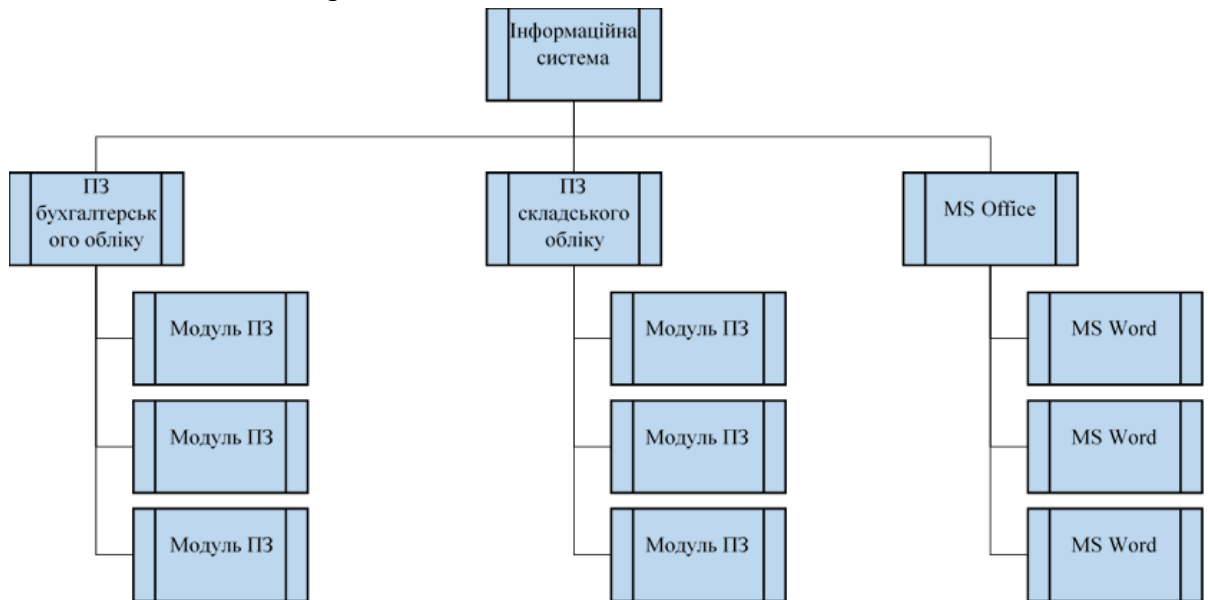


Рис. 14.6 Інформаційна система

### Етап 4. "Опис функцій підрозділів".

На даному етапі описуються функції (рис. 14.7), які виконуються структурними підрозділами компанії.

За результатами опису функцій будується функціональна модель компанії, структурована по організаційній структурі.

На цьому етапі будується функціональна модель об'єкту інформатизації та визначаються підрозділи і ролі виконавців, які відповідальні за виконання кожної з функцій. Це дає можливість перевірити повноту опису і їх відповідність функціям автоматизованих робочих місць (АРМ-ів).

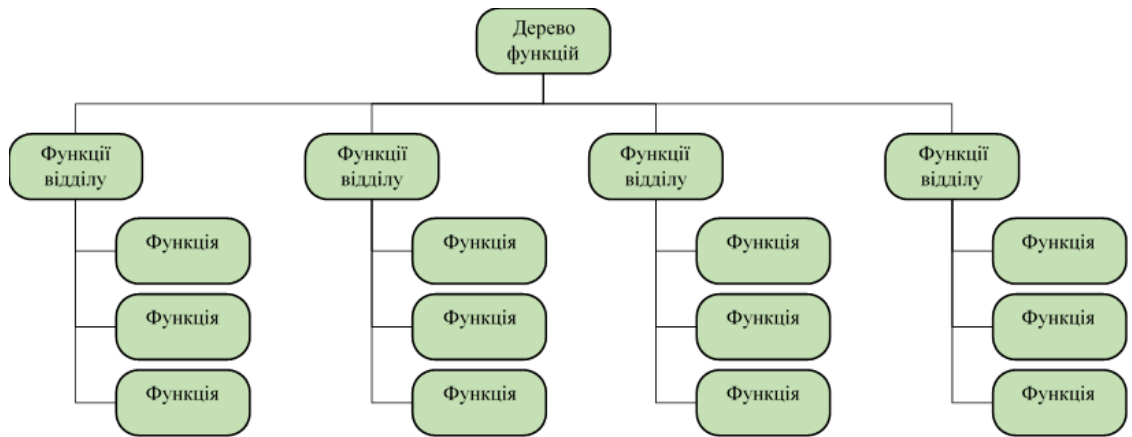


Рис. 14.7 Функції підрозділів

Етап 5. "Опис продуктів і послуг".

На даному етапі описуються продукти та послуги, які виробляє і реалізує компанія.

При побудові моделі продуктів і послуг на верхньому рівні виділяються продуктові групи, які деталізуються на продукти чи послуги (рис. 14.8), що мають різну технологію виробництва, виконання і надання.

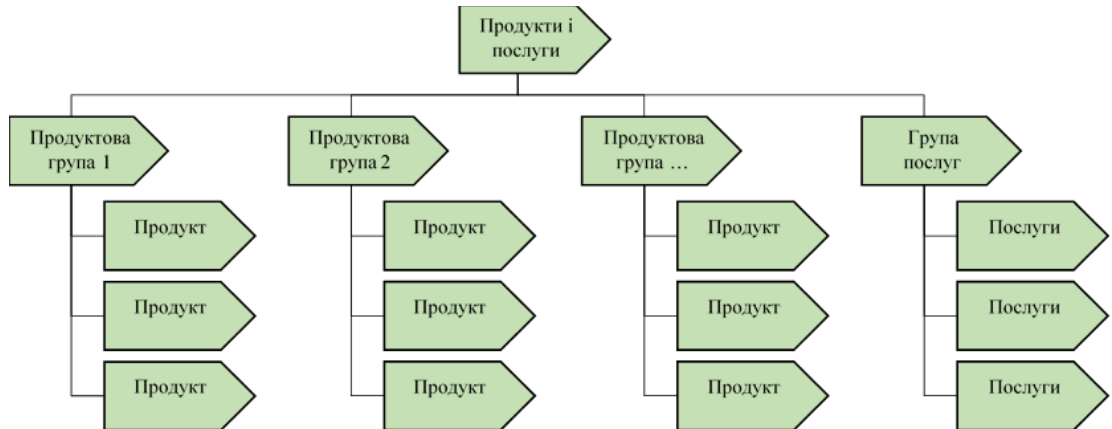


Рис. 14.8 Продукти і послуги

Етап 6. "Виділення та побудова дерева бізнес-процесів".

На даному етапі на основі описаних продуктів і послуг компанії, а також описаних функцій підрозділів виділяються бізнес-процеси, які класифікуються на:

- основні бізнес-процеси;
- бізнес-процеси, що забезпечують задачі та функції управління.

Після проведення класифікації бізнес-процесів будується модель бізнес-процесів компанії (рис. 14.9). Це необхідно, щоб визначити послідовність розробки програмних засобів в залежності від їх пріоритетності.

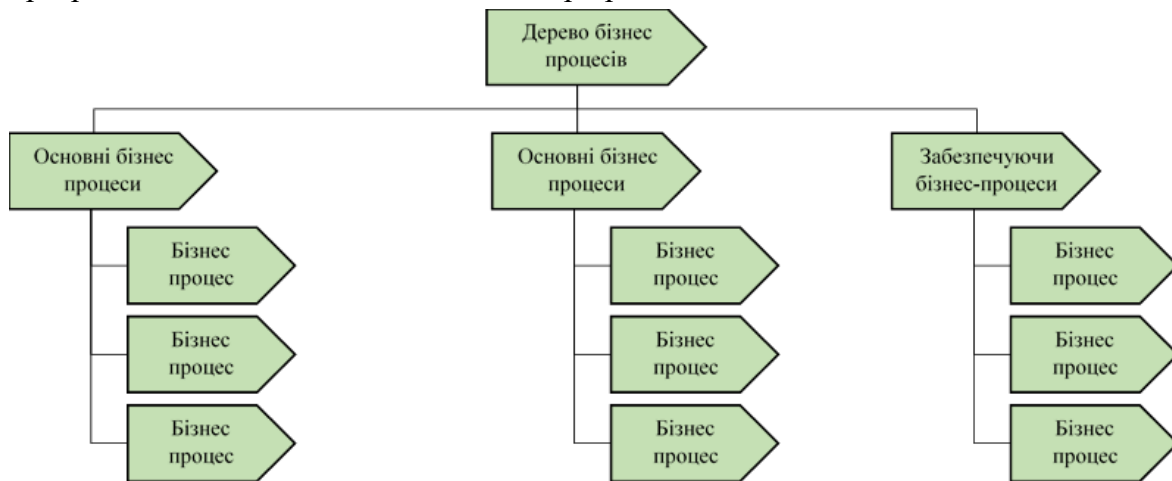


Рис. 14.9 Дерево бізнес-процесів

Етап 7. "Оцінка і вибір бізнес-процесів для подальшого опису, аналізу та оптимізації (реінжинірингу)".

На даному етапі на основі розробленого переліку бізнес-процесів компанії виділяються пріоритетні бізнес-процеси для подальшого опису, аналізу та оптимізації. Вибір пріоритетних бізнес-процесів проводиться на основі наступних критеріїв:

- важливість бізнес-процесу;
- проблемність бізнес-процесу;
- витрати, необхідні для оптимізації бізнес-процесу та подальшої інформатизації.

Для кожного з виділених бізнес-процесів розраховуються кількісні показники відповідно до наведених критеріїв, після чого проводиться їх ранжування.

Для ранжування бізнес-процесів на основі критеріїв важливості і проблемності застосовується матриця "Важливість-проблемність" (рис. 14.10).

Цю роботу виконують, як правило, системні аналітики разом з представниками Замовника інформатизації та майбутніх Користувачів програмних засобів.

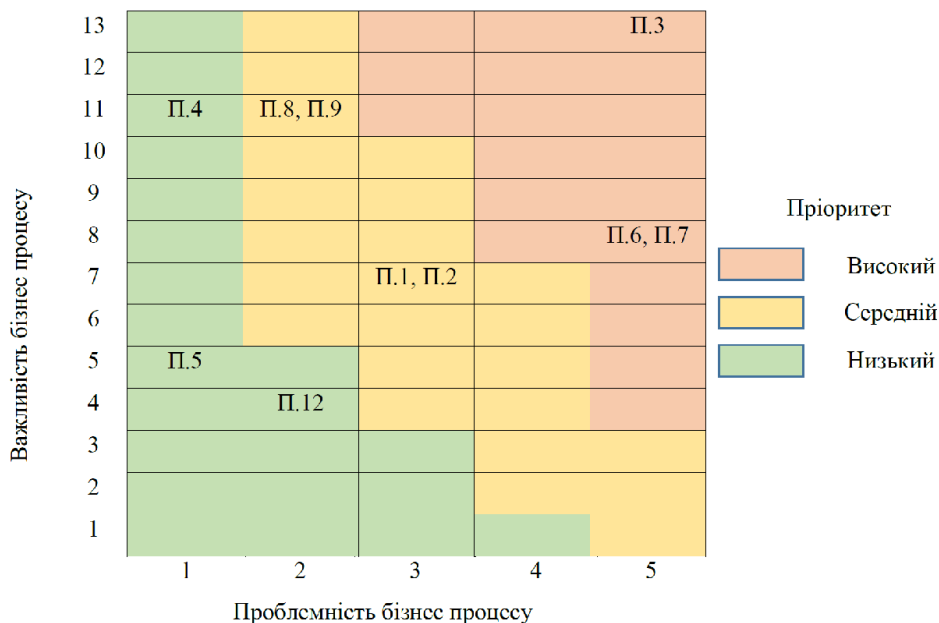


Рис. 14.10 Ранжування бізнес-процесів

Етап 8. "Опис інформаційних потоків для обраних бізнес-процесів".

На даному етапі ідентифікуються інформаційні потоки (рис. 14.11), які беруть участь в описуваних бізнес-процесах, які являють собою інформацію розміщену на паперових носіях та в електронному вигляді.

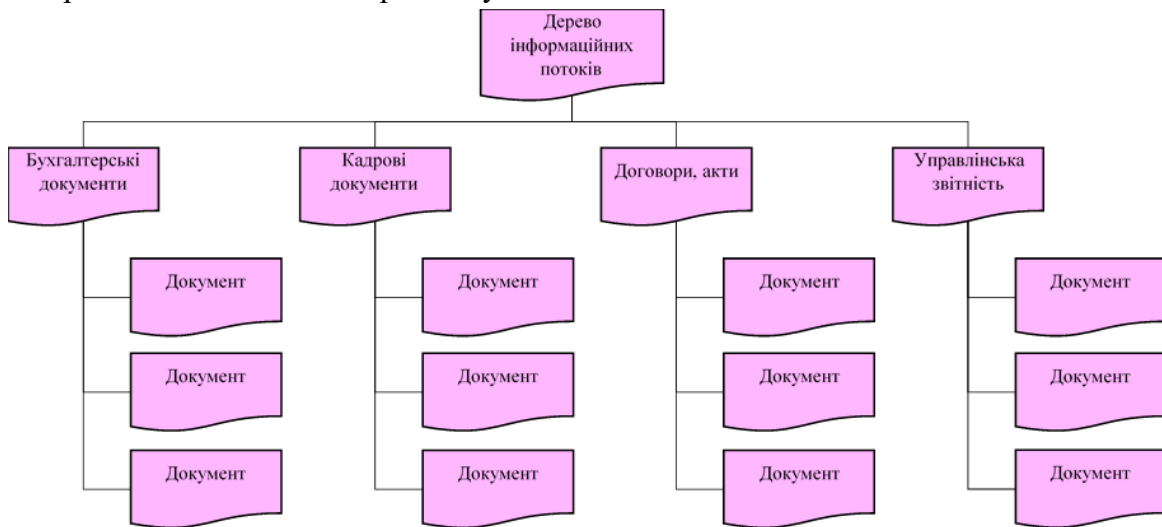


Рис. 14.11 Інформаційні потоки для обраних бізнес-процесів

Дані інформаційні потоки класифікуються, після чого будується модель інформаційних потоків для кожного бізнес-процесу.

Це один з найбільш відповідальних етапів, що визначає структуру баз даних системи та елементи підсистеми документообігу та контролю виконання доручень на об'єкті інформатизації.

Етап 9. "Опис матеріальних потоків, обраних бізнес-процесів".

На даному етапі аналогічно попередньому ідентифікуються матеріальні потоки (рис. 14.12), які беруть участь в описуваних бізнес-процесах. Дані матеріальні потоки класифікуються, після чого будується модель матеріальних потоків для кожного з бізнес-процесу.

Це необхідно для перевірки відповідності матеріальним потокам виявлених вже інформаційних потоків та при необхідності їх доробка (як матеріальних так і інформаційних).

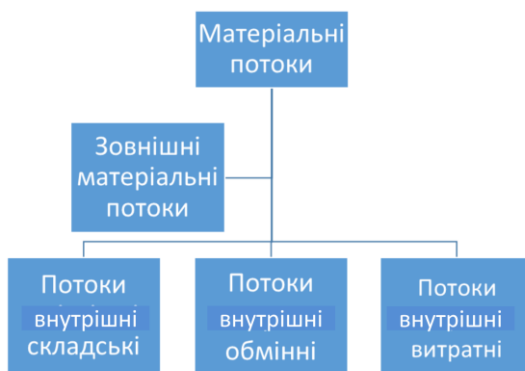


Рис. 14.12 Матеріальні потоки «обраних» бізнес-процесів

Етап 10. "Опис обраних бізнес-процесів".

На даному етапі будуються моделі обраних бізнес-процесів.

Для побудови моделей бізнес-процесів використовуються результати попередніх етапів, а саме:

Використовувані результати.

1 Етап. Опис організаційної структури.

Елементи організаційної структури: відділи, посади.

2 Етап. Опис фінансової структури

Елементи фінансової структури - центри фінансової відповідальності.

3 Етап. Опис інформаційної системи.

Елементи інформаційної системи: модулі, програмне забезпечення.

4 Етап. Опис функцій підрозділів.

Функції підрозділів і функціональні обов'язки працівників.

5 Етап. Опис продуктів і послуг.

Елементи і перелік продуктів і послуг.

8 Етап. Опис інформаційних потоків, обраних бізнес-процесів.  
Інформаційні потоки і види документів.

9 Етап. Опис матеріальних потоків, обраних бізнес-процесів.  
Матеріальні потоки (зовнішні і внутрішні).

Структура моделі бізнес-процесу, розробленої на основі результатів етапів 1-9 (рис. 14.13).

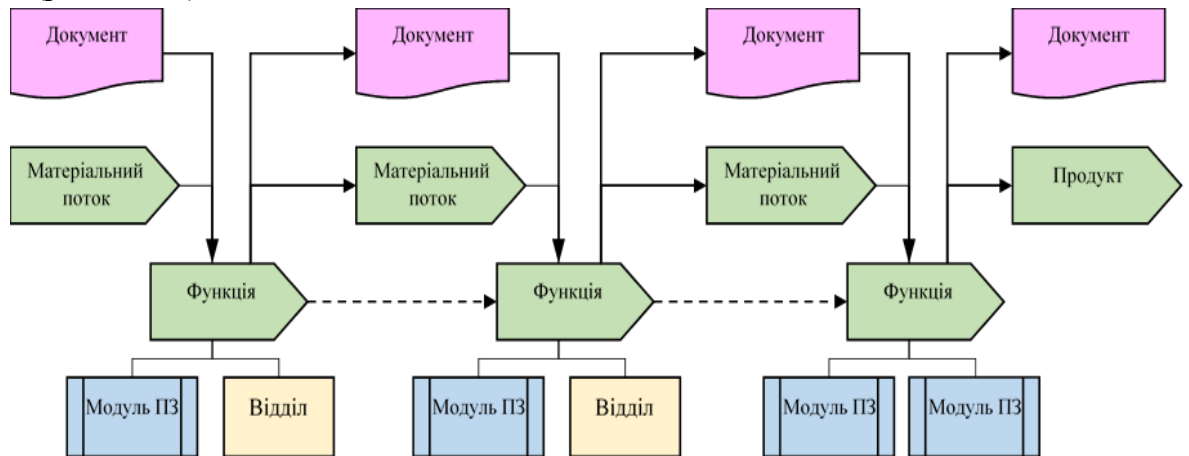


Рис. 14.13 Структура моделі бізнес-процесу після 1-9 етапів

Етап 11. "Збір форм документів та встановлення зв'язків зі схемами бізнес-процесів та інформаційних потоків".

На даному етапі збираються існуючі, які використовуються в компанії, та визначаються додаткові нові форми документів. Після чого вони пов'язуються до відповідних елементів моделей інформаційних потоків і моделей бізнес-процесів.

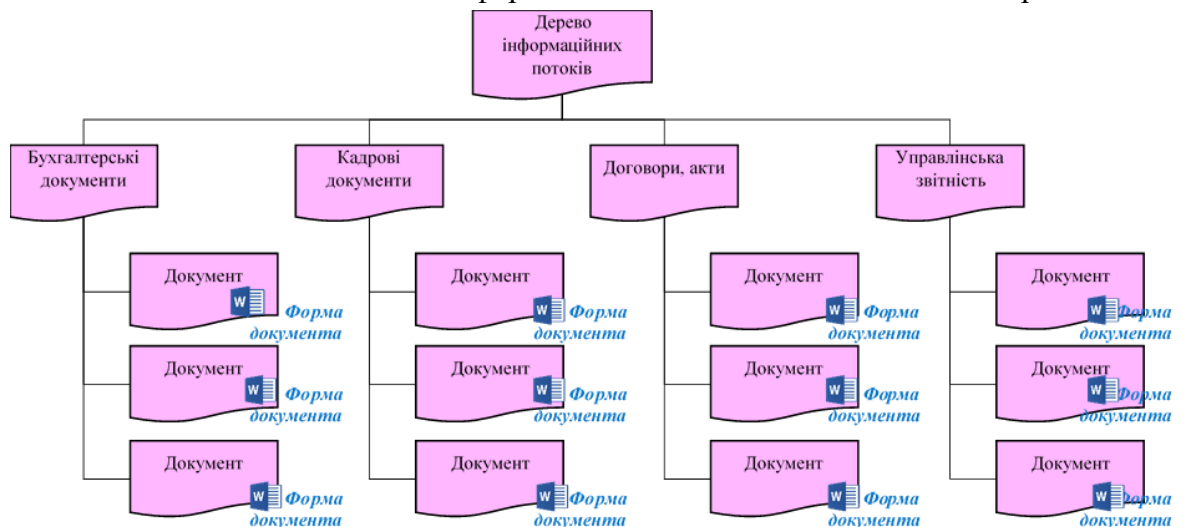


Рис. 14.14 Модель інформаційних потоків з прив'язаними до них документами

«Лінкування» розуміється як процес встановлення взаємозв'язків між документами, які є невід'ємною частиною інформаційного забезпечення процесів на підприємстві, та елементами інформаційних потоків бізнес процесів, які мають бути оптимізовані та автоматизовані в результаті побудови чи реінжинірингу ERP системи об'єкту інформатизації чи її елементів.

Модель інформаційних потоків з прив'язаними до них формами документів зображено на рис. 14.14, а модель бізнес-процесу з "пролікованими" формами документів зображено на рис. 14.15.

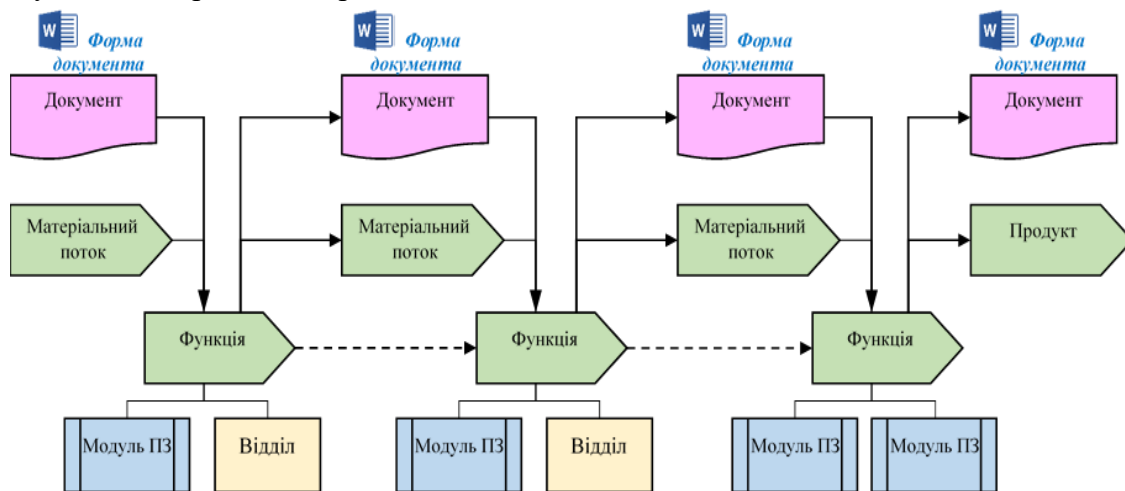


Рис. 14.15 Модель бізнес-процесу з "пролікованими" формами документів

Така узагальнена схема бізнес-процесів з пов'язаними документами, БД та іншими джерелами і споживачами інформації, дозволяє ефективно використовувати об'єктно-орієнтоване проектування (UML) при побудові чи реінжинірингу інформаційно-управлінських автоматизованих систем для підприємств і організацій.

#### Питання до розділу 14

1. Яке призначення ERP-систем?
2. Яке місце займає ERP-система в ієрархії інформаційних систем?
3. Які основні компоненти ERP-систем?
4. Які етапи включає класична технологія опису бізнес-діяльності підприємств та організацій?
5. Для чого використовується ранжування бізнес-процесів?
6. Для чого використовується лінкування у моделі бізнес-процесів?

## РОЗДІЛ 15. МЕТОДОЛОГІЇ МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ ТА СЦЕНАРІЇВ ІНФОРМАЦІЙНИХ СИСТЕМ

### 15.1. Методології моделювання. Типи моделей

В основі різних методологій моделювання предметної області ІС лежать принципи послідовної деталізації абстрактних категорій [1-18].

Зазвичай моделі будуються на трьох рівнях:

- зовнішньому (визначення вимог – що має роботи?);
- концептуальному (специфікації вимог - як має робити?);
- внутрішньому (реалізації вимог – за допомогою чого робити?).

На зовнішньому рівні моделювання визначає що має робити система.

Визначається склад основних компонентів системи: об'єктів, функцій, подій, організаційних одиниць, технічних засобів.

На концептуальному рівні модель відповідає питанням як має функціонувати система.

Визначається характер взаємодії компонентів системи (Концептуальна модель).

На внутрішньому рівні модель відповідає питанням: за допомогою яких алгоритмів та програмно-технічних засобів реалізуються вимоги до системи?

Визначається склад та методи реалізації взаємодії програмно-технічних засобів (рис. 15.1, 18.1, 18.2).

Розглянемо особливості побудови моделей предметної області на трьох рівнях деталізації.

Зовнішній рівень:

Що має робити система (рис. 15.1)?

Концептуальний рівень:

Як має функціонувати система (рис. 18.1)?

Внутрішній рівень:

За допомогою яких алгоритмів та програмно-технічних засобів реалізуються вимоги до систем (рис. 18.2)?

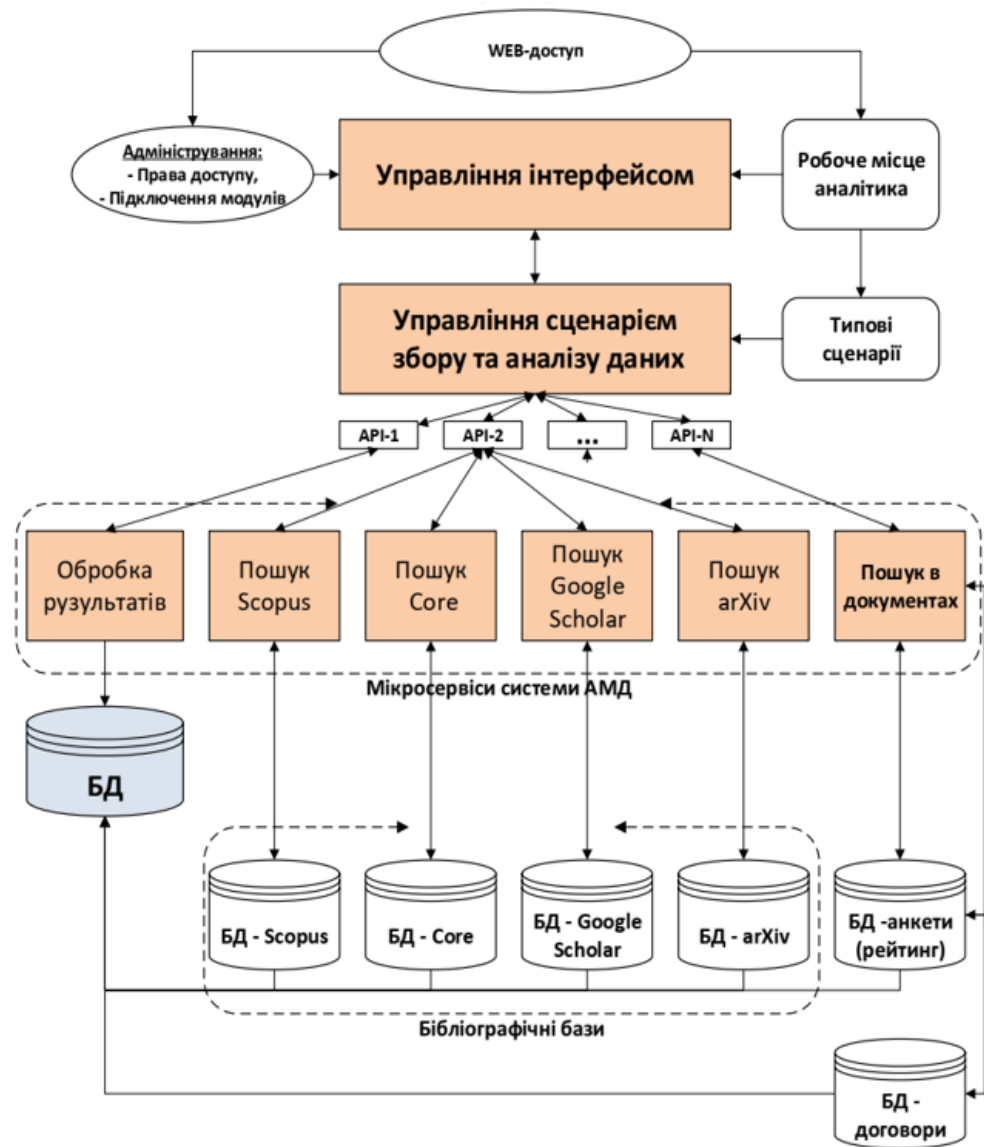


Рис. 15.1 Структура пілотного проекту системи аналізу міжнародної діяльності (АМД)

## 15.2. Функціональна структура

Функція (операція) є перетворювачем вхідних параметрів елементів об'єктів у вихідні.

Послідовність взаємозалежних входів і виходів функцій становить бізнес-процес. Функція бізнес-процесу може породжувати результати будь-якої природи (матеріальні, фінансові, інформаційні).

Сценарії у функціональній структурі розглядаються як:

- послідовність взаємозалежних входів і виходів функцій, які відносяться до однієї ролі (чи виконавця) в рамках бізнес-процесу, що розглядається;
- або виконання однієї з можливих послідовностей дій в рамках одного з екземплярів бізнес-процесу.

При цьому бізнес-процеси та інформаційні процеси, зазвичай, нерозривні.

Тобто функції бізнес-процесу неспроможні здійснюватися без інформаційної підтримки.

Наприклад, відвантаження готової продукції здійснюються на основі документа "Замовлення", який, у свою чергу, породжує документ "Накладна", що супроводжує партію відвантаженого товару.

Функція може бути представлена однією дією або деякою сукупністю дій.

В останньому випадку кожній функції може відповідати певний процес, в якому можуть існувати свої підпроцеси, і так далі, доки кожна з підфункцій не представлятиме деяку неподільну послідовність дій.

На зовнішньому рівні моделювання визначається перелік основних бізнес-функцій чи видів бізнес-процесів.

Загальний БП може включати перелік інших бізнес-процесів нижнього рівня.

На концептуальному рівні виділені функції декомпонуються та на їх основі будуються ієрархії взаємопов'язаних функцій.

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

### **15.3. Структура управління**

У сукупності функцій бізнес-процесу можливі альтернативні чи циклічні послідовності залежно від умов протікання процесу. Ці умови пов'язані з подіями, що відбуваються в зовнішньому середовищі або в самих процесах і з утворенням певних станів об'єктів (наприклад, замовлення: прийняте, відкинута, відправлене на коригування).

Події викликають виконання функцій, які, у свою чергу, змінюють стан об'єктів і формують нові події, і так далі, поки не буде завершено деякий бізнес-процес. Тоді послідовність подій складає конкретну реалізацію бізнес-процесу відповідно до певних умов.

Це розглядається як окремий сценарій реалізації бізнес-процесу.

Таким чином події викликають виконання того чи іншого сценарію дій, як одного з альтернативних шляхів реалізації екземпляру бізнес-процесу.

Кожна подія описується з двох точок зору: інформаційної та процедурної.

Інформаційно подія відображається у вигляді деякого повідомлення, що фіксує факт виконання деякої функції зміни стану або появи нового стану.

Процедурно подія викликає виконання нової функції, тому для кожного стану об'єкта повинні бути задані описи цих викликів.

На зовнішньому рівні визначаються список зовнішніх подій, що викликаються взаємодією підприємства (об'єкта) із зовнішнім середовищем (платежі податків, відсотків за кредитами, постачання за контрактами тощо), та список цільових установок, яким мають відповідати бізнес-процеси (регламент виконання процесів, підтримка рівня матеріальних запасів, рівень якості продукції та ін.).

На концептуальному рівні встановлюються бізнес-правила, що визначають умови використання функцій у разі виникнення подій та досягненні станів об'єктів.

На внутрішньому рівні виконується формалізація бізнес-правил як тригерів, що визначають виклики, необхідних для реалізації бізнес-процесу, програмних модулів.

#### **15.4. Технічна структура**

Топологія визначає територіальне розміщення технічних засобів у структурних підрозділах підприємства, а комунікація – технічний спосіб реалізації взаємодії структурних підрозділів.

На зовнішньому рівні моделі визначаються типи технічних засобів обробки даних та їх розміщення за структурними підрозділами.

На концептуальному рівні визначається спосіб комунікацій між технічними комплексами структурних підрозділів: фізичне переміщення документів, машинних носіїв, обмін інформацією каналами зв'язку тощо.

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

Описані моделі предметної області націлені на проектування окремих компонентів ІС:

- даних;
- функціональних програмних модулів;
- управляючих програмних модулів;
- програмних модулів інтерфейсів користувачів;
- структури технічного комплексу.

Для якісного проектування зазначених компонентів потрібна побудова моделей вищого рівня, що пов'язують різні моделі між собою.

У найпростішому випадку як такі моделі взаємодії можуть використовуватися матриці перехресних посилань: «об'єкти – функції», «функції – події», «організаційні одиниці – функції», «організаційні одиниці – об'єкти», «організаційні одиниці – технічні засоби» тощо. Такі матриці не наочні та не відображають особливості реалізації взаємодій.

### **15.5. Структурний аналіз**

Для правильного відображення взаємодій компонентів ІС важливо здійснювати їхнє спільне моделювання, відображаючи об'єкти та функції, властиві їм. Методологія структурного системного аналізу суттєво допомагає у вирішенні таких завдань.

Структурним аналізом прийнято називати метод дослідження інформаційних систем, яке починається з її загального огляду, а потім деталізується, набуваючи ієрархічної структури з дедалі більшим числом рівнів.

Для таких методів характерно:

- розбиття на рівні абстракції з обмеженою кількістю елементів (від 3 до 7);
- обмежений контекст, що включає лише суттєві деталі кожного рівня;
- використання строгих формальних правил запису;
- послідовне наближення результату.

Структурний аналіз ґрунтується на двох базових принципах - принципі розділення та ієрархічної впорядкованості.

Побудова складних за структурою об'єктів реалізується шляхом їх розбиття на безліч менших незалежних об'єктів (так званих «чорних ящиків») та організація цих об'єктів у деревоподібні ієрархічні структури значно підвищують розуміння складних систем.

### **15.6. Основні компоненти моделі структурного аналізу**

В моделі структурного аналізу розглядають наступні основні компоненти:

1. Операція – елементарна (неподільна) дія, що виконується на одному робочому місці.
2. Функція – сукупність операцій, згрупованих за певною ознакою.
3. Бізнес-процес – пов'язана сукупність функцій, під час виконання якої споживаються певні ресурси і створюється товар (предмет, послуга, наукове відкриття, ідея), що становить цінність споживача.

4. Підпроцес – це бізнес-процес, що є структурним елементом бізнес-процесу вищого рівня і представляє цінність для споживача.

5. Бізнес-модель – структурований графічний опис мережі процесів та операцій, пов'язаних із даними, документами, організаційними одиницями та іншими об'єктами, що відображають існуючу або передбачувану діяльність підприємства.

Процес бізнес-моделювання може бути реалізований в рамках різних видів методики, що відрізняються, перш за все, своїм підходом до того, що є модельована організація, як об'єкт інформатизації.

Методики прийнято ділити на об'єктні та функціональні (структурні).

Об'єктні методики розглядають організацію, що моделюється, як набір взаємодіючих об'єктів – виробничих одиниць.

Об'єкт визначається як незалежна від опису реальність (!) реальність – предмет або явище, що мають поведінку, яка чітко визначається.

Метою застосування об'єктної методики є виділення об'єктів, що становлять організацію, та розподіл між ними відповідальності за виконувани дії.

Функціональні методи, найбільш відомою з яких є методика IDEF, розглядають організацію як набір функцій, що перетворює потік інформації, що надходить, у вихідний потік.

Процес перетворення інформації споживає певні ресурси.

Основна відмінність функціональної методики від об'єктної методики полягає у чіткому відокремленні функцій (методів обробки даних) від самих даних.

З погляду бізнес-моделювання, кожен із представлених підходів має свої переваги.

Об'єктний підхід дозволяє побудувати більш стійку до змін систему, що 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 на основі функцій інтуїтивно краще розуміється користувачами інформаційної системи при отриманні від них інформації про їх поточну роботу.

### **15.7. Інжиніринговий підхід**

Найбільш ефективним для цілей проектування ІС є інжиніринговий підхід до формування моделей підприємства, як об'єкта інформатизації.

Організаційний аналіз підприємства за такого підходу проводиться за певною схемою, спрямованою на отримання повної бізнес-моделі підприємства.

Підприємство розглядається як відкрита, соціально-економічна система, що належить ієрархічній сукупності відкритих зовнішніх систем (ринок, держава та ін.) та внутрішніх підсистем (відділи, цехи, бригади та ін.) і має певні цілі.

Можливості підприємства визначаються характеристиками її структурних підрозділів та організацією внутрішніх та зовнішніх зв'язків.

Такий підхід реалізується, як правило, при розробці програмних засобів ERP (Enterprise Resource Planning) систем.

Побудова бізнес-моделі починається з опису взаємодії підприємства із зовнішнім середовищем, на основі якого формулюються його місія та цілі (рис 15.2).



Рис. 15.2 Основні компоненти бізнес-моделі

Місія відображає два аспекти:

- з одного боку — це діяльність, яку проводить підприємство для того, щоб виконати функцію, для якої воно було засноване, – надання замовникам продукту або послуги;
- з іншого – це механізм, за допомогою якого підприємство реалізує свої цілі та завдання (наприклад отримати дохід > 1 млн.\$).

Місія відбиває головні елементи та направленість реалізації задач та методів інформатизації підприємства.

Вона вказує необхідну для замовника послідовність реалізації задач інформатизації та їх важність для замовника.

Це дає в свою чергу можливість більш ефективно вирішити задачі по заключенню контракту на виконання робіт з інформатизації з замовником.

Таким чином, місія є основною діловою концепцією підприємства, викладеною у вигляді ряду положень, які визначають його взаємини з іншими суб'єктами:

- що отримує замовник щодо задоволення своїх потреб;
- хто, для чого і як може виступати партнером підприємства;
- на якій основі передбачається будувати відносини з конкурентами (яка, зокрема, готовність піти на тимчасові компроміси);
- що отримують власник та акціонери від бізнесу;
- що отримують від бізнесу менеджери підприємства;
- що отримує від підприємства персонал;
- як будуватимуться відносини підприємства з державою (зокрема, можлива участь у підтримці державних програм).

### **15.8. Бізнес-потенціал підприємства**

Бізнес-потенціал підприємства – це набір видів діяльності, спрямований на задоволення потреб конкретних сегментів ринку.

На основі опису бізнес-потенціалу формується узагальнене уявлення про організаційну структуру – визначаються центри комерційної відповідальності та склад основних ресурсів, необхідних для виробництва товарів та послуг.

Бізнес-потенціал дозволяє визначити функціонал підприємства – перелік основних бізнес-функцій, функцій менеджменту та функцій забезпечення, які потрібні для підтримки всіх видів діяльності.

Бізнес-потенціал дозволяє визначити найбільш важливі бізнес-процеси підприємства, основні ролі у бізнес-процесах підприємства та найбільш важливі сценарії, як рольові так і сценарії дій.

### **Питання до розділу 15**

1. В чому суть трьох рівнів побудови моделей?
2. Що визначає функціональна структура?
3. В чому суть структури управління?
4. Які елементи включає технічна структура?
5. На чому ґрунтується структурний аналіз?
6. Які основні компоненти моделі структурного аналізу?
7. В чому суть інжинірингового підходу?

## РОЗДІЛ 16. ПОМИЛКИ У ОПИСІ БІЗНЕС-ПРОЦЕСІВ СКЛАДНИХ ВИРОБНИЧИХ ОБ'ЄКТІВ ІНФОРМАТИЗАЦІЇ

### 16.1. Типи критичних помилок у бізнес-процесах

При описі моделей бізнес-процесів виникають критичні типові помилки [1, 17]. Які не тільки ускладнюють розуміння та особливості бізнес-процесів, але і призводять до значних помилок при створенні програмних систем, які побудовані на основі моделей цих бізнес-процесів.

Серед таких типів помилок особливо розповсюджені наступні:

- очікування;
- незавершене закінчення;
- марні операції;
- неправильний порядок операцій;
- зайві переміщення;
- дублювання операцій;
- відхилення.

### 16.2. Очікування

В певних ситуаціях не можливо виконати дії в процесі доти, доки не відбудеться певна попередня подія. А оскільки подія є результатом дії, навіть якщо ми її не бачимо, така ситуація утворює очікування (рис 16.1). Тобто, якщо розвиток процесу залежить від події, яка є наслідком іншого процесу, або операції всередині процесу, формується очікування.

Пошук втрат, типу "очікування" відбувається через пошук подій, настання яких залежить від дій виконавця. Наприклад: завершення дії програми, настання певної дати, інші дії людини.

У наведеному нижче прикладі, адміністратор не може скласти звіт, доки не отримані дані. При цьому дані будуть отримані не відразу після запиту. З погляду управління бізнес-процесами, така ситуація є очікуванням і є потенційною втратою, оскільки знижує безперервність процесу.

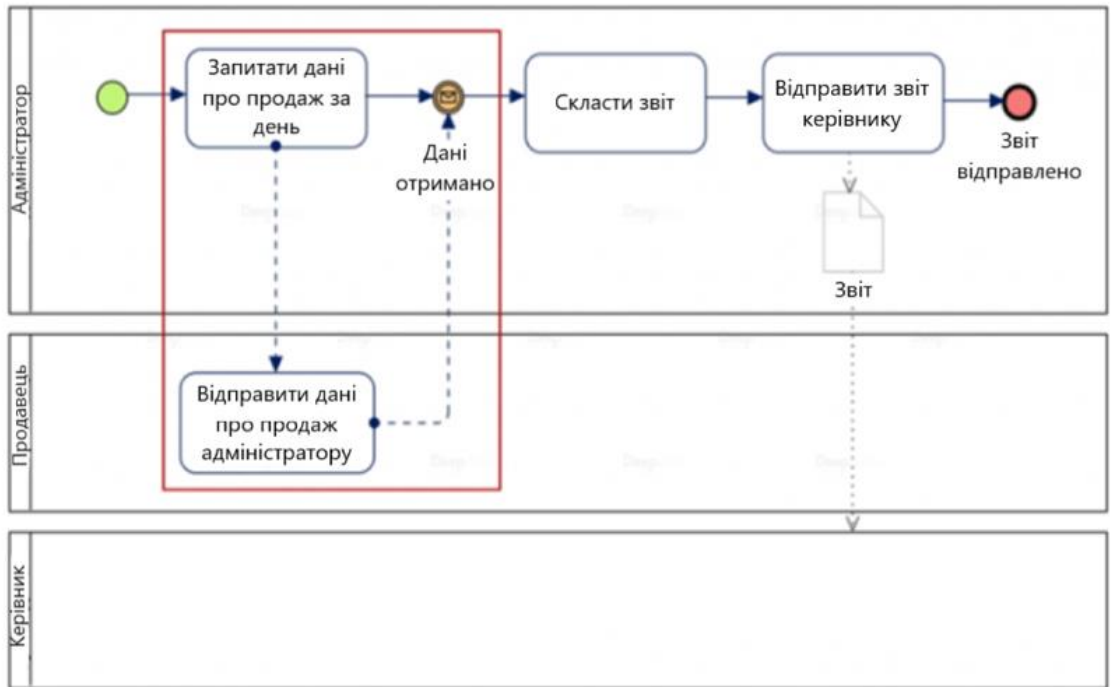


Рис. 16.1 Очікування в процесі

### 16.3. Незавершене закінчення

Кожен сценарій процесу повинен мати логічне завершення – результат, який буде використано далі. Однак, буває і так, що дії виконані, ресурси витрачені, а результат нікому не потрібний. Незавершене закінчення (рис 16.2) з'являється тоді, коли ви не можете відповісти на запитання, що робити далі.

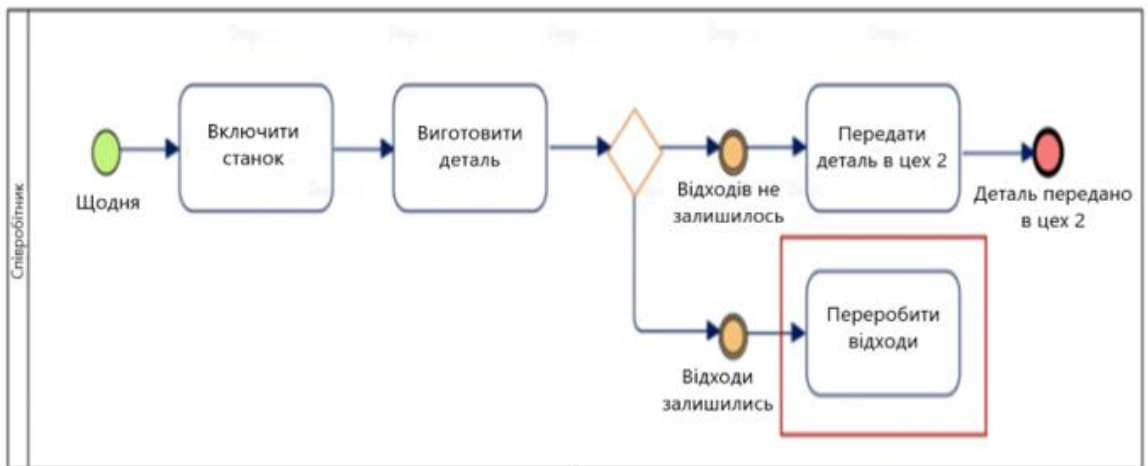


Рис. 16.2 Незавершене закінчення в процесі

Щоб знайти незавершене закінчення, необхідно перевірити всі сценарії на наявність подій закінчення. Якщо ви знайшли сценарій без завершення, ви знайшли закінчення без завершення.

Приклад. При виготовленні деталі, можуть залишитись відходи і співробітник виконує їх переробку. Але, що відбувається з переробленими відходами? Куди процес рухається далі? Якщо відповідь на це питання відсутня то ситуація потенційно вказує на незавершене закінчення.

## 16.4. Марні операції

Марні операції (рис. 16.3) – операції, що не мають користі, тобто не створюють цінності ні для клієнта, ні для компанії. Наприклад: формальне узгодження. Коли учасник процесу узгоджує підписом процеси що не мають відношення до виконання призначеного процесу.

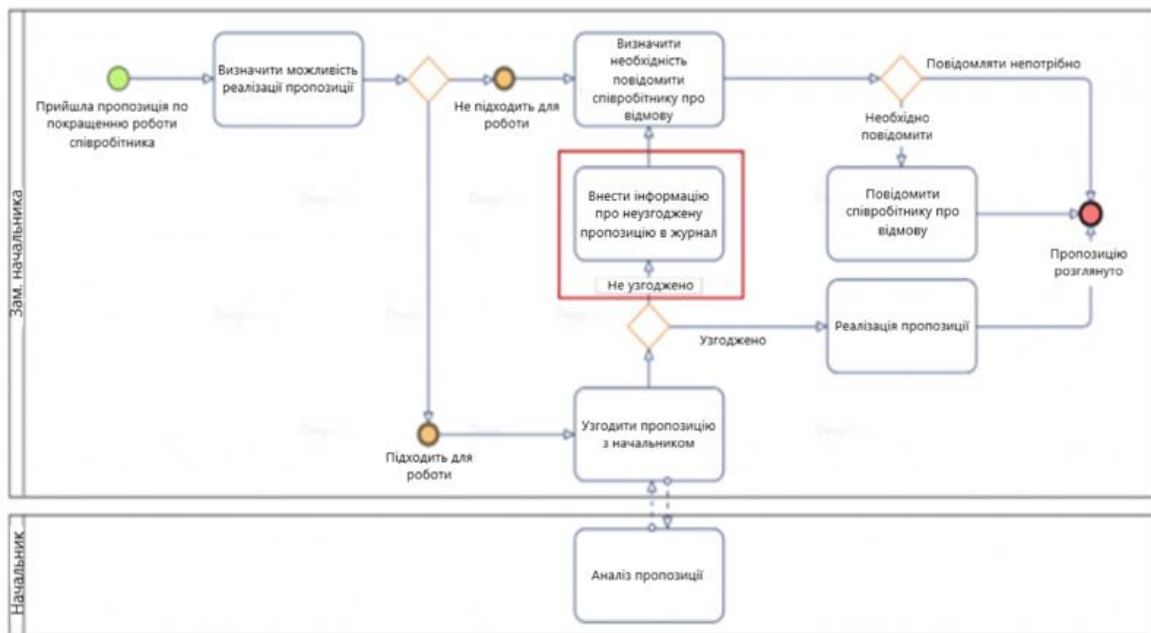


Рис. 16.3 Марна операція

Для пошуку марної операції, доведеться уважно переглянути всі операції та до кожної поставити наступні питання:

- чи приносить операція користь зовнішньому клієнту?
- чи приносить операція користь будь-якому внутрішньому клієнту?
- чи приносить операція користь компанії?

Якщо на всі питання отримано відповідь «НІ», значить операція є марною. Так, пошук втрат може бути дуже простим. В прикладі нижче, до уваги потрапила операція «Внести інформацію про не узгоджені пропозиції в журнал».

Можливо, це не очевидно, але вказана операція не приносить користі зовнішнім і внутрішнім клієнтам, та й компанії від неї користі немає. У такому разі операція може вважатися марною і необхідно розглянути можливість її виключення з процесу.

## 16.5. Неправильний порядок операцій

Іноді, з метою більш ефективної роботи процесу, потрібно поміняти операції місцями. Від того, де стоїть операція, може залежати тривалість та вартість процесу. Спробуйте змінити порядок виконання операцій (рис 16.4). Якщо одна з комбінацій призводить до зменшення тривалості або вартості процесу, ймовірно ви знайшли можливість зробити процес ефективнішим.

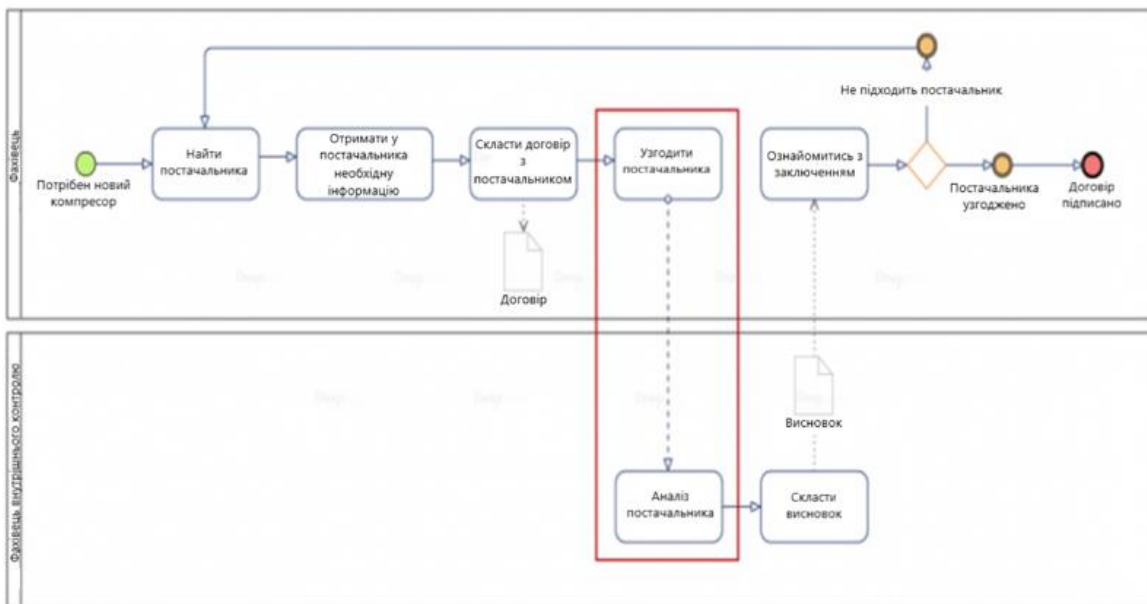


Рис. 16.4 Неправильний порядок операцій в процесі

Наприклад. Звичайний процес узгодження закупівлі. Для покупки необхідно знайти постачальника, скласти договір та узгодити його зі спеціалістом внутрішнього контролю.

На діаграмі видно, що узгодження поставника відбувається після з'ясування інформації та укладання договору. Якщо спеціаліст внутрішнього контролю не погодить постачальника, то попередня робота була виконана дарма. Доведеться

повернутися на початок. Цього не станеться, якщо узгодити постачальника відразу, після вибору.

## 16.6. Зайві переміщення

Чим менш переміщення (рис 16.5) в процесі, тим краще. Кожне переміщення збільшує ризик ушкодження, затримки та ін. На переміщення втрачаються ресурси.

Щоб знайти зайві, необхідно визначити всі місця в процесі де виконуються переміщення. Це може бути передача документів, пересування товарів тощо. Тому потрібно з'ясувати, хто є початковим відправником, а хто кінцевим адресатом. Потім визначити, чи є проміжні пересування. Якщо так, то оцінити їх. Якщо переміщень можливо уникнути, то вони є зайвими.

Наприклад. Співробітник створює документ та відправляє його начальнику відділу. Начальник відділу пересилає документ керівнику підрозділу.

Навіщо в даному начальник відділу? Співробітник може безпосередньо направити документ керівнику підрозділу. Типовий приклад зайвого переміщення відображено на рис. 16.5.

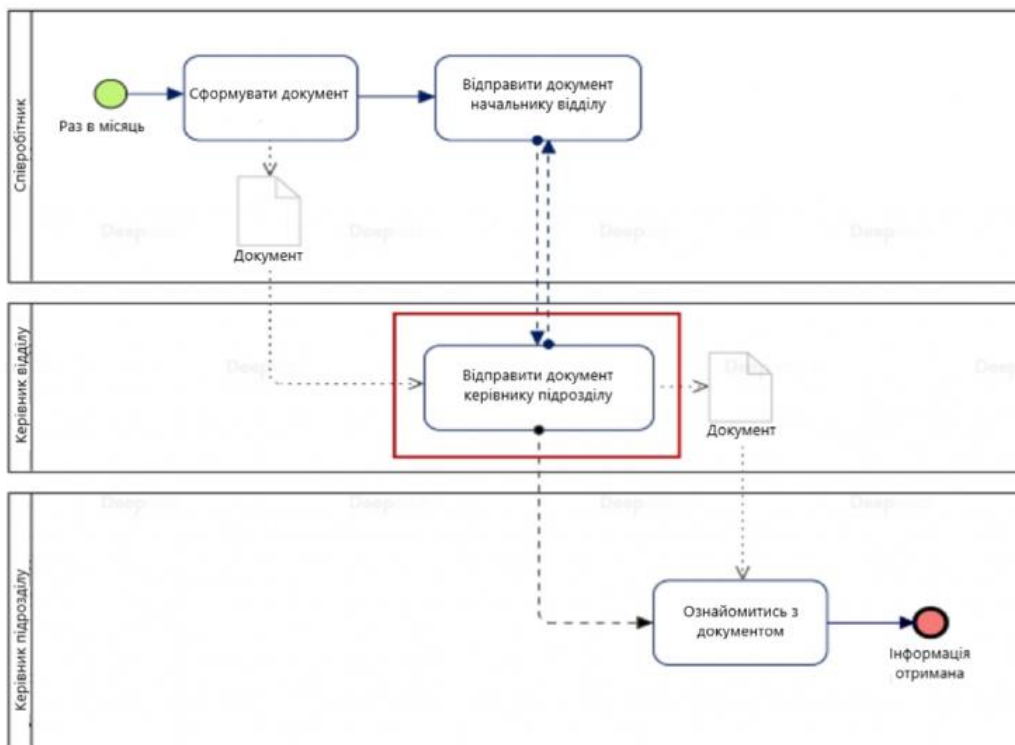


Рис. 16.5 Зайве переміщення

## 16.7. Дублювання операцій

Дуже часто, пошук втрат зводиться до пошуку дублюючих операцій. Дублювання (рис 16.6) виникає, коли операція виконується декілька раз в різних місцях. Причому в операцій може бути різні назви, їх можуть виконувати різні співробітники, але такі операції матимуть спільну мету і зміст.

Для того щоб відшукати дублювання необхідно порівняти операції - одну з іншою. Якщо операції виконуються заради одного й того ж результату, то скоріш за все вони є дублюючими.

Наприклад, в процесі «Контроль виконання стандартів обслуговування», контролер повторно виконує одну й ту ж операцію: «Виконати аналітичну довідку на ресурс» та «Передача аналітичної довідки зам. начальнику, особисто».

Зам. начальника може зайти на корпоративний ресурс та побачити довідку, навіщо передавати її особисто?

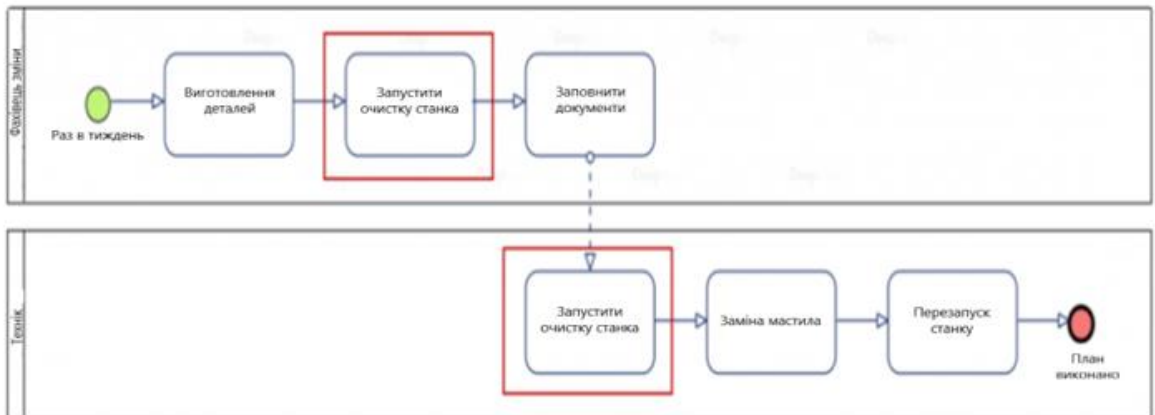


Рис. 16.6 Дублювання операцій

## 16.8. Відхилення

При виконанні процесу можуть виникати ситуації пов'язані з втратами - несправності, помилки, недоробки і т.ін. Це ситуації, які перешкоджають виконанню процесу. Вони виникають у разі: низької кваліфікації працівника, поганої роботи обладнання, неякісних матеріалів тощо.

Відхилення (рис. 16.7, 16.8) – порушення нормальної роботи процесу. Пошук втрат такого типу може бути дуже складним. Щоб знайти відхилення у процесі, необхідно знайти події, які формують відгалуження від основних сценаріїв. Тобто вони формують сценарії, в яких частина операцій не має відношення до одержання

продуктів процесу. Можна сміливо сказати, що це події, які створюють «проблеми» процесу.

Наприклад, несправність принтера. Конструкція типу «компенсація» також може бути гарним індикатором потенційного відхилення.

В цьому прикладі, є подія на межі операцій «Не вказано номер телефону». Ця подія є відхиленням, так як породжує додаткові, небажані операції для процесу, і для його усунення знадобляться компенсаційна дія «Зв'язатися з клієнтом електронною поштою».

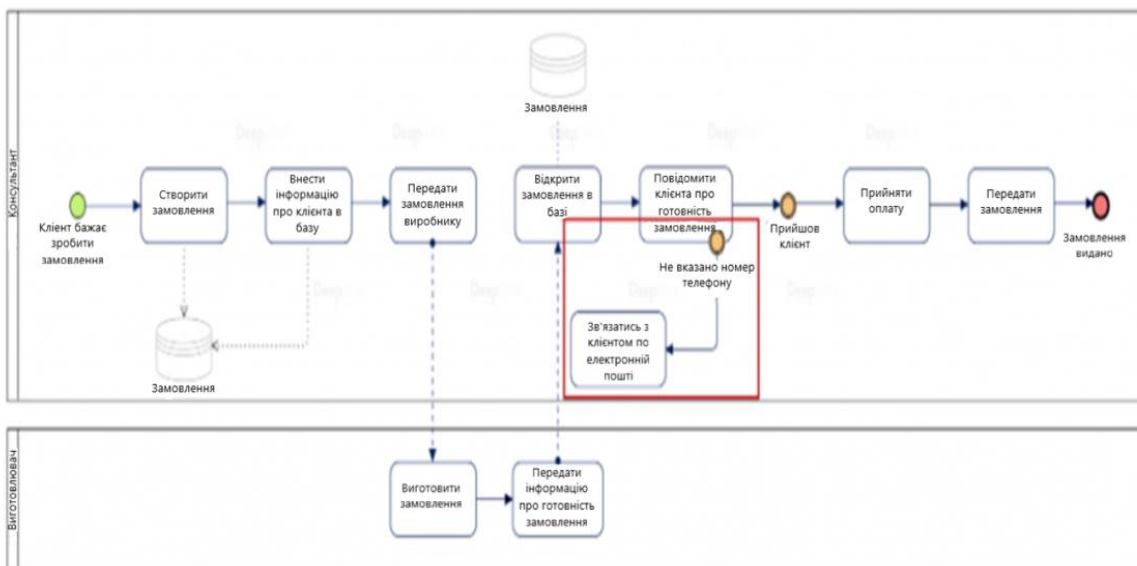


Рис. 16.7 Відхилення в процесі. Приклад 1

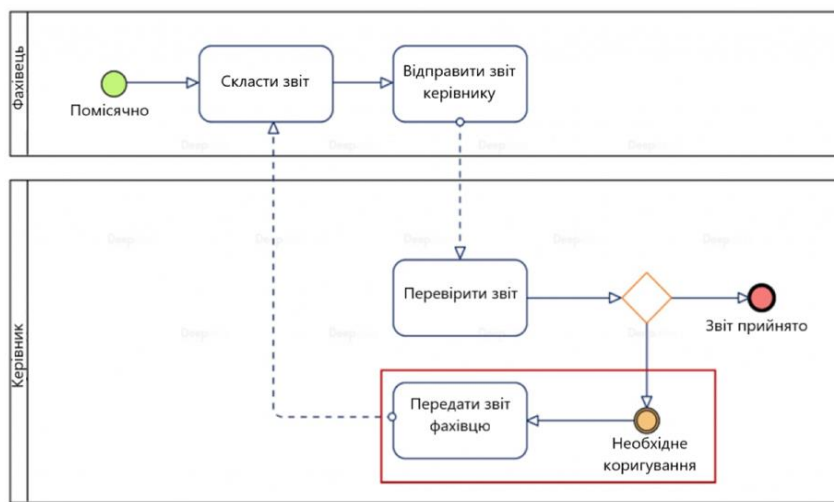


Рис. 16.8 Відхилення в процесі. Приклад 2

Ще приклад. Фахівець помісячно готує звіт та відправляє керівнику на перевірку. Керівник перевіряє та повертає на коригування за необхідністю.

В процесі присутнє розгалуження. В ньому два варіанти «Звіт прийнято» та «Необхідне коригування». Подія «Необхідне коригування» заважає завершити процес, вона повертає процес назад. В такій ситуації, це відхилення.

Пошук втрат необхідно виконувати періодично, згідно прийнятим процедурам та прийнятому плану. Намагайтесь зменшувати час виконання процесів, кількість марних операцій, відхилень. Прибирайте повтори та зайві переміщення.

Ці прості приклади допоможуть вам винайти втрати, без складних аналітичних підходів, лише застосуванням діаграм процесів.

### **Питання до розділу 16**

1. Які типи критичних помилок у описі бізнес-процесів?
2. В чому суть помилки типу «очікування»?
3. Чим характерна помилка «незавершене закінчення»?
4. Чим відрізняються помилки типу «марні операції» та «неправильний порядок операцій»?
5. В чому особливість помилки «зайві переміщення»?
6. В чому суть помилки типу «дублювання операцій»?
7. Чим характерна помилка «відхилення в процесі»?

## РОЗДІЛ 17. МЕТОДИКА ФОРМУВАННЯ СЦЕНАРІЮ АНАЛІТИЧНОЇ ДІЯЛЬНОСТІ

### 17.1. Аналітична діяльність

Аналітична діяльність у різних галузях визначається як:

- робота по пошуку та збору інформації;
- аналіз та глибоке осмислення;
- прийняттям рішень щодо результатів аналізу в тій чи іншій ситуації;
- тематична обробка інформації;
- візуалізація аналітичних звітів;
- верифікація;
- отримання управлінських рішень на базі нових знань.

Метою аналітичної діяльності є забезпечення інформаційних потреб організації та підтримка прийняття рішень.

Такі задачі для більшості складних динамічних систем (такі як ланка промисловості, бюджет країни, розвиток напрямку у технічних науках та інше) приводять до необхідності визначення для кожної з задач свого особливого шляху роботи з великими даними (Big Data).

### 17.2. Процес аналітичної діяльності

Процес аналітичної діяльності може бути представлений у вигляді ітераційної послідовності ряду етапів збору та обробки інформації, сукупний опис яких дає уявлення про нього у цілому:

- аналіз інформації про предметну область;
- первинний аналіз інформації, що надходить до інформаційно-аналітичної системи (ІАС), та її структуризація;
- порівняльний аналіз інформаційних ресурсів ІАС;
- алгоритмічний аналіз інформаційних ресурсів ІАС;
- інтелектуальний (семантичний) аналіз інформаційних ресурсів ІАС;
- планування та контроль виконання заходів щодо забезпечення розвитку процесів предметної області ІАС.

### **17.3. Сценарій аналітичної діяльності (САД)**

Під сценарієм аналітичної діяльності (САД) будемо розуміти певну послідовність дій, яку виконує аналітик або група аналітиків при вирішенні інформаційно-аналітичних задач [2, 9, 11, 12].

Побудова сценаріїв є методикою реалізації системного дослідження щодо оцінки функціонування складних динамічних систем.

Особлива увага приділяється цілям системи: їхньому виникненню, формулюванню, деталізації, декомпозиції, структуризації, аналізу та іншим питанням визначення мети.

Може бути використана така класифікація типів САД:

- організаційно-технічний;
- інформаційно-технічний;
- інформаційно-аналітичний;
- системно-аналітичний.

### **17.4. Модель сценарію аналітичної діяльності**

Структуру САД можна одержати шляхом послідовної декомпозиції САД до рівня мікрооперацій (МО) (таблиця 17.1).

На нижньому рівні деталізації виділяються МО, що відповідають найменш деталізованим елементам аналітичної діяльності, які можна відобразити у моделі.

Можна виділити такі елементи САД (рис. 17.1):

- мета виконання сценарію;
- завдання (програма) виконання сценарію;
- інформаційні ресурси;
- програмно-технічні ресурси;
- початок (ініціалізація) сад;
- послідовність виконання операцій (власне сценарій);
- аналітичний результат виконання сад.



Рис. 17.1 Модель сценарію аналітичної діяльності

Виконавці сценарію – ролі.

Основні ролі, що використовуються при формуванні САД:

- керівник;
- керівник групи;
- аналітик;
- експерт;
- технік;
- секретар;
- виконавці.

Слід виділити 2 види необхідності чи причини початку САД:

- планова;
- ініціалізована у зв'язку із виникненням проблемної ситуації (аварії на інфраструктурних об'єктах, стихійні явища та інші).

### 17.5. Різновиди реалізації САД

Спосіб організації САД:

- локальний;
- розподілений.

Таблиця 17.1 Типові операції в корпоративній інформаційно-аналітичній системі

№	Код	Назва	Зміст
1	ЗД	Збір інформації	Запит, прийом каналами зв'язку, розміщення в сховищі
2	ПОД	Первинна обробка даних	Вибірка, групування, оформлення таблиць, графіків
3	АД	Аналіз даних	Статистична обробка, побудова залежностей, розрахунок прогнозних значень
4	ЕД	Експертиза даних	Перевірка відсутності протиріч ухваленим фактам
5	ІД	Інтерпретація даних	Виконання аналітичного дослідження та оформлення звітнього документа
6	ПЛ	Планування	Складання, розгляд та оформлення планів
7	ЗВ	Звітність	Складання, розгляд та оформлення звітів
8	К	Контроль	Контроль виконання планів, звітів, розпоряджень, вказівок, документів
9	ВСП	Вироблення складних рішень	Вивчення проблеми, організація колегіальної роботи, прийняття рішень
10	ПД	Підготовка документів	Розробка концепції складного документа, підготовка проекту, узгодження, розсилання адресатам

Суб'єкт виконання САД:

- індивідуальний;
- груповий;
- кооперативний;
- масовий.

Об'єкт виконання САД:

- обробка документа;
- інформування керівництва;

- консультування підлеглих;
- колегіальне обговорення.

Можливість формалізації:

- операції САД не підлягають формалізації;
- частково формалізуються;
- повністю формалізуються.

Вид результату:

- надання вихідних даних;
- формування концепції рішення;
- розробка проекту рішення;
- експертне заключення;
- участь у колективному рішенні;
- редагування вихідного документа.

САД можна визначити як:

$$S = \{I, Z, O, R, A, T\},$$

де:

- I – ініціатор операції;
- Z – суб'єкт операції;
- O – об'єкт аналізу;
- R – отримувач результату;
- A – порядок виконання операції;
- R – результат операції;
- T – термін виконання операції.

Кожен САД можна розкласти на ряд мікрооперацій (елементарних операцій), які мають такі особливості:

- чітку визначеність постановки завдання, методу її розв'язання;
- наявність технологічних альтернатив рішення;
- ситуаційність використання.

Приклади МО:

- запитати інформацію;
- надати інформацію;
- запропонувати варіант рішення;
- проаналізувати варіант рішення;
- порівняти інформаційні об'єкти (джерела інформації);

- порівняти з тенденцією в минулому (визначити тенденції релевантності, ефективності ...);
- екстраполювати тенденцію розвитку подій;
- прогнозувати тенденцію розвитку подій на віддалену перспективу;
- оцінити наслідки прийняття рішення;
- оцінити ризик;
- відкласти ухвалення рішення;
- повідомити рішення.

### **17.6. Середовища функціонування САД**

Функціонування САД передбачає наявність певного середовища, яке включає:

- лінгвістичне забезпечення;
- інформаційне забезпечення;
- апаратне забезпечення;
- програмне забезпечення;
- нормативно-правове забезпечення;
- організаційне забезпечення.

### **17.7. Методика формування сценарію аналітичної діяльності**

Послідовність формування сценаріїв:

- аналіз існуючих елементів КІАС;
- опис завдань аналітичної діяльності;
- визначення виконавців САД;
- побудова списку окремих операцій;
- призначення виконавців для кожного САД;
- завдання для всіх операцій основних параметрів (тривалість, термін виконання);
- визначення взаємної підпорядкованості входження окремих операцій САД;
- визначення залежностей попередження операцій (логічних залежностей);
- призначення ресурсів для виконання САД;
- моделювання виконання САД;
- аналіз результатів моделювання;
- у разі потреби повернення до першого кроку.

## 17.8. Підсистеми САД

Підсистема збору даних (рис 17.2) забезпечує отримання даних із різних джерел у формі: доповіді, повідомлення, аналітичного огляду, звіту, довідки, відомості, телеграми, телефонограми.

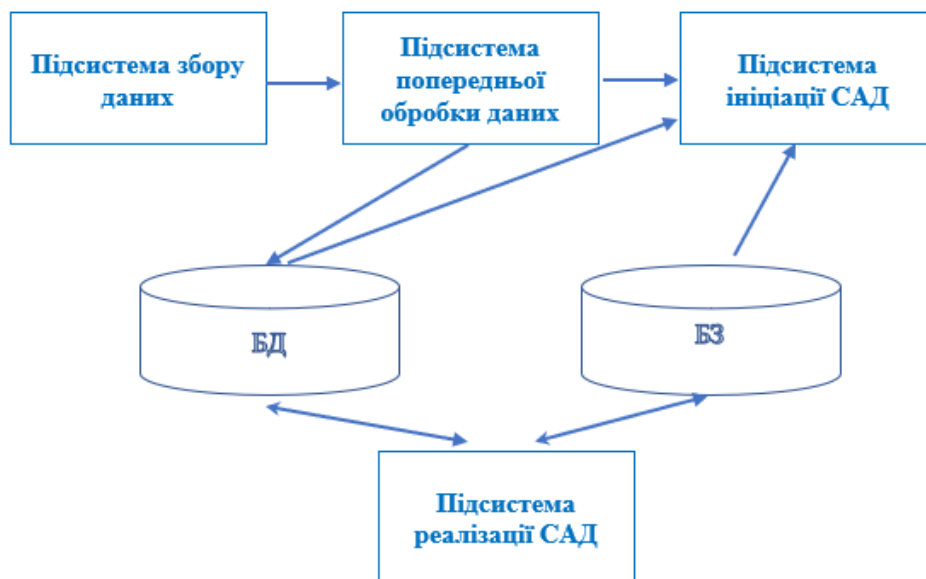


Рис. 17.2 Структура середовища формування САД

Підсистема попередньої обробки даних будується на базі лінгвістичного процесора (ЛП), що виконує роль посередника між оператором та масивом даних, з якого необхідно отримати потрібну для активації САД інформацію. Завданням ЛП є перетворення отриманих даних на набір семантичних структур, є формальним відображенням «змісту» отриманої інформації. Мета такого перетворення — забезпечити вихідні дані для роботи під час запуску САД та наповнення БД.

Підсистема запуску САД забезпечує обробку вхідної інформації, інформації з БД та бази знань та здійснює запуск САД. Системи ініціалізації САД можна класифікувати за методами аналізу вхідної інформації, періодичністю роботи, реакцією, централізацією.

Підсистема реалізації САД забезпечує всі необхідні засоби для виконання САД.

## 17.9. Процес реалізації САД

Процес реалізації САД можна розглядати як ряд окремих, але пов'язаних між собою процесів:

- процес запуску (ініціації чи активації) САД;
- процес планування — визначення та удосконалення цілей та вибір найкращого з альтернативних планів дій, що ведуть до досягнення цілей САД;
- процес виконання - координація використання людських та інших ресурсів до виконання САД;
- процес контролю - регулярний моніторинг та оцінка ходу виконання робіт для визначення відхилень від плану, за потреби, коригуючих дій;
- процес завершення - формальне прийняття результатів САД та оформлення типового САД.

### **17.10. Модель сценарію розвитку ситуації**

Сценарій можна описати як сукупність наступних елементів:

- операції, які виконуються в процесі виконання (кроки сценарію);
- зміст дій учасників на кожному кроці сценарію;
- ресурси (учасники сценарію, задіяні бази даних і бази знань, технічні та програмні);
- опис умов виконання та обмежень операцій сценарію;
- посилання на ділянки проектної документації, що роз'яснюють зміст кроку сценарію;
- посилання на екранні форми, що ілюструють зміст дій учасників сценарію;
- коментарі.

### **17.11. Використання графів**

Для побудови сценарію використовується граф (рис. 17.3), вершини якого відображають різні етапи виконання функціонального завдання, а дуги відповідають заданим послідовностям взаємодій.

Для побудови графа сценарію пропонується використання абстрактного автомату, що описує вимоги на підставі опису аналітичної задачі.

Процес, у якому переходи виконуються під впливом зовнішніх дій, моделюється за допомогою автомата.

Використовуючи автомати, можна моделювати багато різноманітних процесів. Також можна дослідити питання вибору напрямку розвитку сценарію в залежності від ситуацій, що виникають в ході рішення задач аналітичної діяльності.

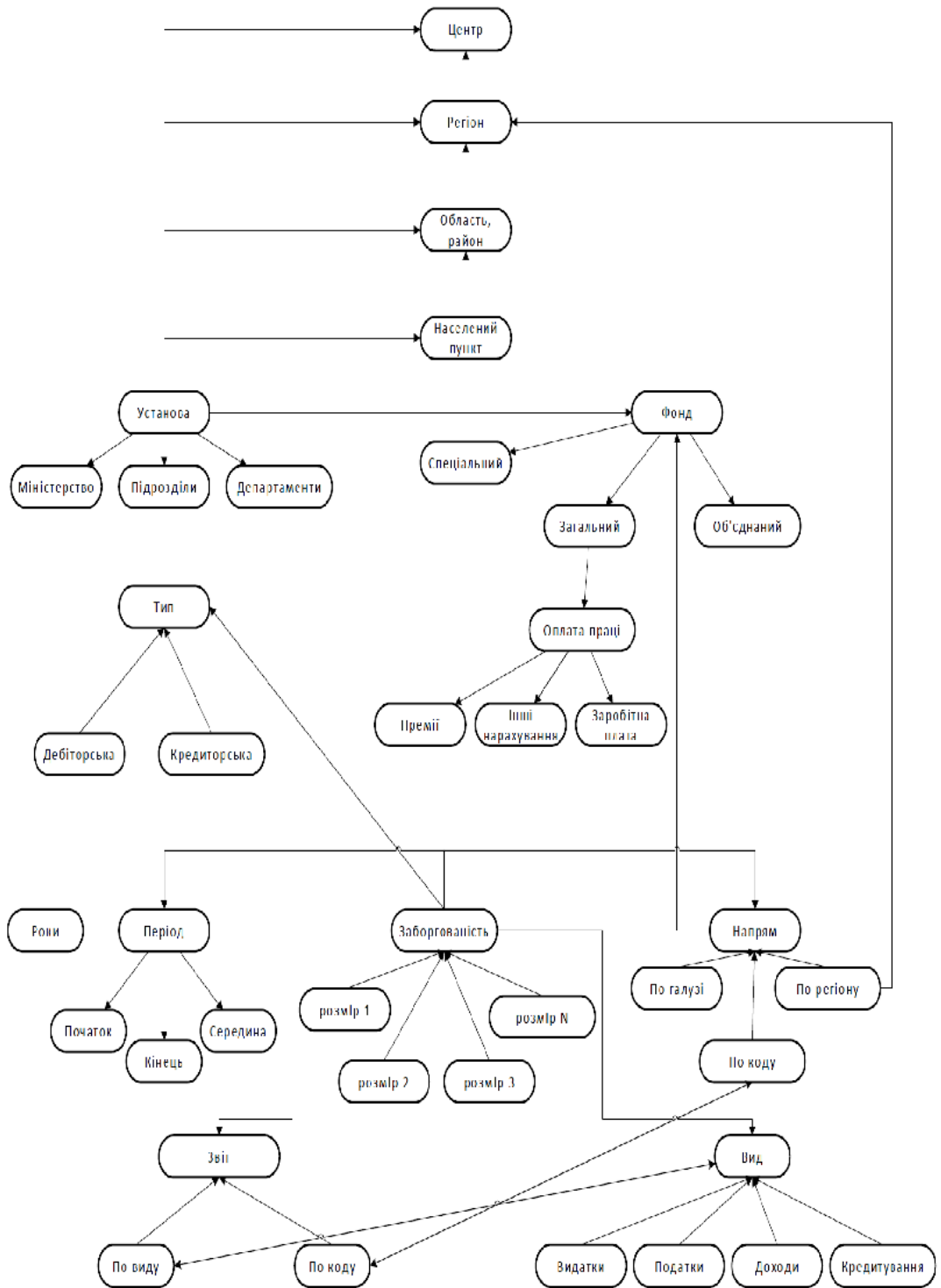


Рис. 17.3 Приклад графу

## **17.12. Технологія моделювання сценаріїв**

Технологія моделювання сценаріїв включає наступні етапи:

- підготовку сценарію до моделювання;
- визначення списку АРМ і їхніх функцій для виконання завдання;
- передачу вимог щодо виконання завдання групі адміністрування та управління моделюючого комплексу;
- конфігурування моделюючого комплексу і передачу постановки задачі функціональним підсистемам для реалізації сценарію;
- відтворення моделюючим комплексом ситуацій;
- моніторинг роботи сценарію;
- реалізацію алгоритмів моделювання та видачу результатів;
- виконання комплексу стратегій;
- збереження результатів моделювання;
- аналіз результатів моделювання;
- складання звітної документації за результатами моделювання.

## **17.13. Застосування сценарного підходу в аналітичній діяльності**

Сценарний підхід широко застосовується при аналізі ризиків, зокрема у сфері інформаційної безпеки при дослідженнях систем управління, моделюванні взаємодії користувачів у розподілених системах, при аналізі та моделюванні надзвичайних ситуацій (рис. 17.4, 17.5).

Сценарій аналізу включає наступні етапи:

- 1) вибір систем інтеграції веб-документів (джерел інформації);
- 2) формування запиту в середовищі обраної системи. Знаходження тематичних публікацій за запитом за допомогою систем контент-моніторингу;
- 3) визначення динаміки тематичних публікацій за запитом;
- 4) визначення критичних точок у динаміці тематичних публікацій;
- 5) визначення основних подій у критичних точках;
- 6) виявлення об'єктів моніторингу;
- 7) виявлення і візуалізація взаємозв'язків;
- 8) прогноз розвитку подій.

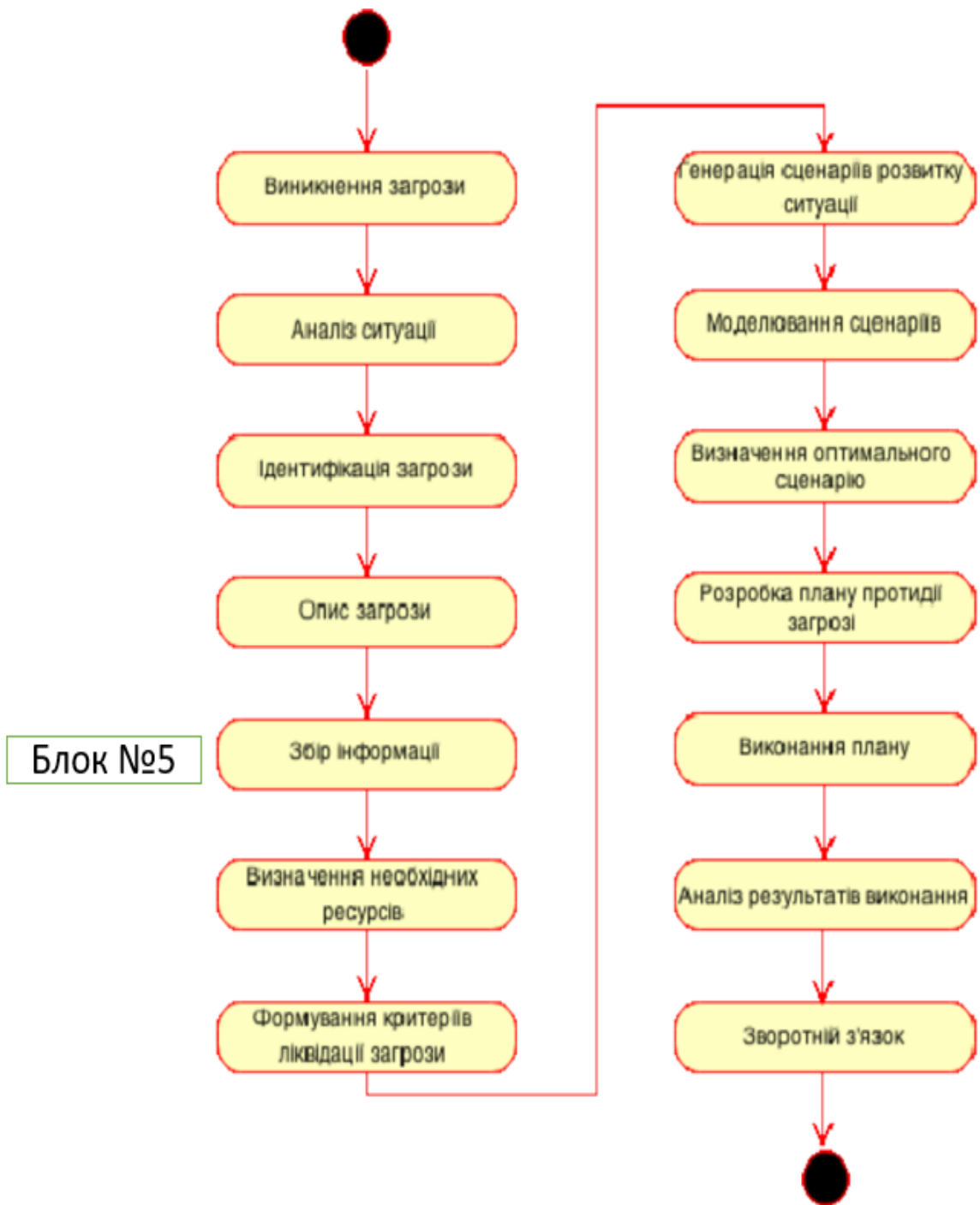


Рис. 17.4 Сценарій протидії інформаційній загрозі

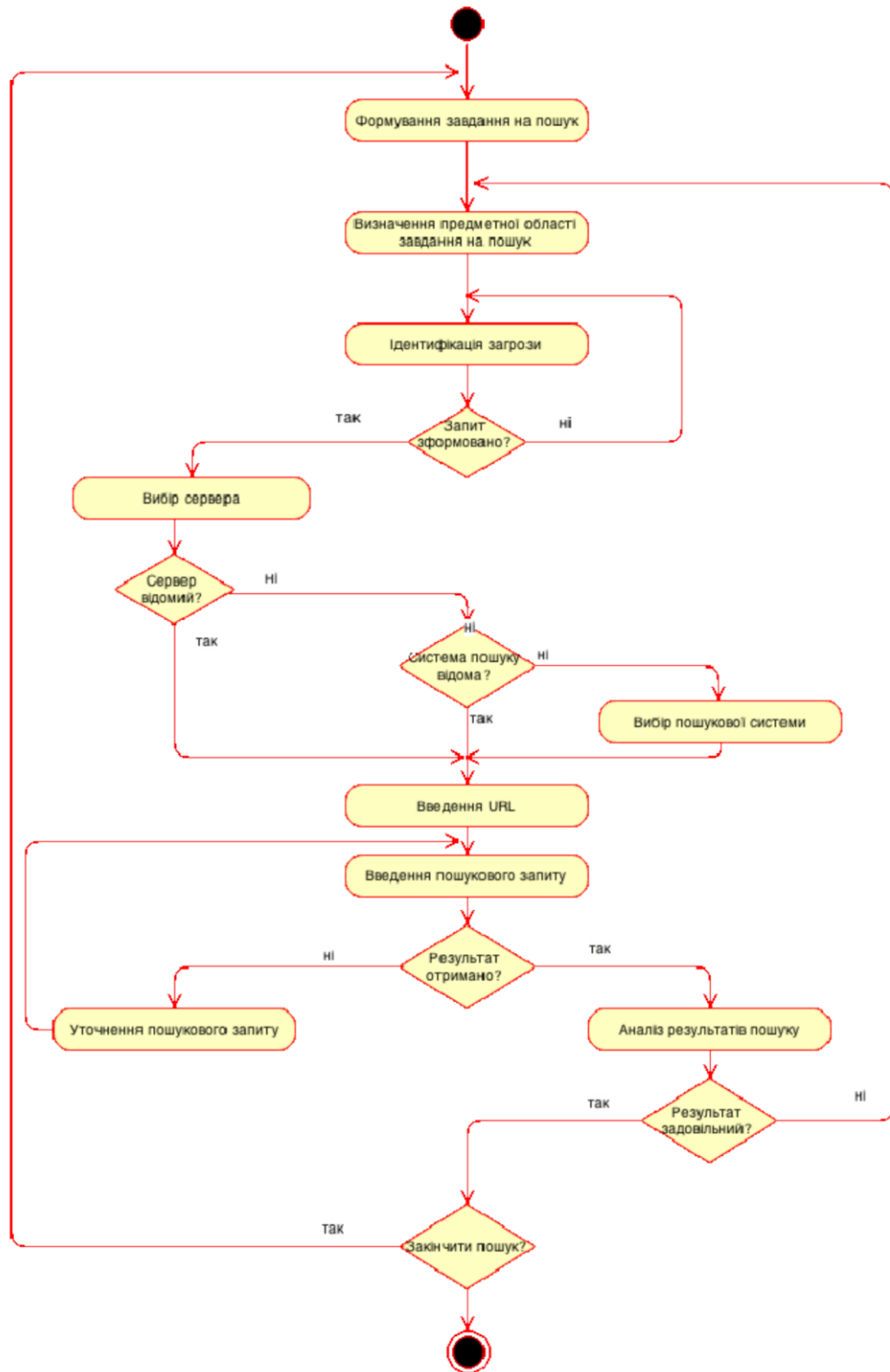


Рис. 17.5 Збір інформації (блок №5)

## **Питання до розділу 17**

1. Як визначається поняття «Аналітична діяльність»?
2. Які є процеси аналітичної діяльності?
3. Що таке сценарій аналітичної діяльності (САД)?
4. С чого складається модель сценарію аналітичної діяльності?
5. Які основні ролі, що використовуються при формуванні САД?
6. Які є типові операції інформаційно-аналітичної системи?
7. Які є підсистеми САД?
8. С чого складається процес реалізації САД?
9. Які можна виділити етапи технології моделювання сценарію?

## **РОЗДІЛ 18. ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ В РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ**

### **18.1. Сценарії збору та аналізу інформації**

Сценарії може розглядатися як спосіб збору та аналізу складного інформаційного середовища інформаційно-аналітичної системи (ІАС), у якому є безліч значимих, взаємопов'язаних та альтернативних інформаційних джерел. При цьому сам сценарій може включати в себе можливий набір подій, що визначає розвиток тих чи інших факторів, які впливають на результат діяльності. Системи збору та первинного аналізу даних є одним з найбільш складних та важливих елементів інформаційно-аналітичних систем.

З точки зору розробки системи збору та первинного аналізу даних під типовим сценарієм пошуку інформації за запитом аналітика може розумітися послідовність дій (кроків) системи пошуку інформації, обраної аналітиком для вирішення такого завдання. Типові сценарії можуть змінюватися і перетворюватися в ході пошуку інформації з подальшим збереженням нових версій і примірників версій. Після проведення аналітичних досліджень, нові примірники версій можуть зберігатися та ставати окремими версіями типових сценаріїв, для подальшого використання аналітиком для вирішення подібних аналітичних задач.

Однією з головних проблем, для системи аналізу міжнародної діяльності, при зборі первинної інформації, є її неповнота та неналежна актуальність у переважній більшості первинних інформаційних джерел. Це призводить до необхідності дублювання, при визначенні джерел інформації, аналізу якості інформації під час вилучення інформації та необхідності побудови складних алгоритмів послідовності виконання процедур аналізу та обробки інформації, основаних на типових сценаріях проведення аналізу інформації за відповідними запитом аналітика.

При цьому, складність таких процедур, вимагає не тільки пристосування структури виконання кроків типових сценаріїв до конкретної задачі, що ставить аналітик, але й пристосування параметрів окремих елементів, що реалізують певні кроки сценарію до цих особливостей, як під час проведення аналізу, так і після його закінчення.

Ефективна реалізація таких складних за структурою та значних за обсягами обробки даних систем можлива тільки з використанням методів та процесів паралельного збору інформації, що можна якісно реалізувати на розгалужені мережі з використанням мікросервісної архітектури.

## **18.2. Мікросервісна архітектура програмних засобів в реалізації сценаріїв**

Мікросервісна архітектура побудови інформаційних систем являє собою комплекс незалежно розгорнутих сервісів. Побудована таким чином система має можливість еволюціонувати по частинах, в першу чергу, за рахунок того, що кожен мікросервіс значною мірою автономний. Крім того, така архітектура дозволяє гнучке масштабування компонентів інформаційної системи, тим самим забезпечуючи оптимальне використання наявного серверного обладнання. При цьому масштабування кожного мікросервісу проводиться незалежно від інших мікросервісів. Стало можливим виділяти обчислювальні ресурси з урахуванням реальної активності користувача і затребуваності тієї чи іншої функціональності системи.

Мікросервісна архітектура – це підхід до створення додатку, що має на увазі відмову від єдиної, монолітної структури. Тобто замість того, щоб виконувати всі обмежені контексти, додатки на сервері за допомогою внутрішніх процесних взаємодій, ми використовуємо кілька невеликих доданків, кожний з яких відповідає якомусь обмеженому контексту. Причому ці програми можуть працювати на різних серверах і взаємодіють один з одним по мережі, наприклад за допомогою HTTP.

Одна з важливих переваг мікросервісів полягає в тому, що ми можемо використовувати різні технології для вирішення однієї і тієї ж задачі. Наприклад, в кожному мікросервісі використовувати різні бібліотеки для парсингу або різні інструменти обробки даних. Це дає можливість побудови адаптивного, до типу виконуваної задачі, програмного забезпечення, яке реалізує певні елементи машинного навчання. Не виключено, що якась кількість технологій і бібліотек вийде з-під контролю. Так що виберіть базовий набір інструментів і звертайтеся до інших тільки тоді, коли це дійсно потрібно.

Крім того, такий підхід дає можливість будувати та використовувати паттерни схем збору та первинної обробки інформації як складові частини сценаріїв пошуку та обробки інформації для конкретних задач оцінки рівня міжнародної діяльності.

На сьогодні існує досить багато підходів та засобів для реалізації мікросервісної архітектури. Серед них найбільш розвинутими та широко використовуваними є платформи Docker та Kubernetes.

## **18.3. Структура програмної реалізація підсистеми збору даних**

Для вирішення такої складної задачі, як розробка підсистеми збору даних, що стає перед розробниками будь-якої інформаційно-аналітичної системи, може бути

використана архітектура, яка побудована на принципах сценарного підходу з використанням адаптивної мікросервісної архітектури з елементами машинного навчання. Така підсистема призначена для паралельного збору даних з різних за структурою, формою зберігання та місць розташування інформаційних джерел у реальному часі. Ця підсистема дозволяє збирати різноманітну інформацію за якими завгодно ознаками. Принципова схема реалізації такої системи представлена на рис. 18.1.

Розглянемо можливості мікросервісної архітектури на прикладі аналізу міжнародної діяльності. Загальний процес роботи інформаційно-аналітичної системи оцінки рівня міжнародної діяльності можна поділити на такі основні кроки:

1. Обробка запиту та вибір чи формування сценарію.
2. Формування на основі сценарію структури мікросервісів для виконання задач щодо запиту.
3. Виконання збору та обробки інформації за запитом.
4. Консолідація та збереження результатів для подальшого аналізу.
5. Аналіз роботи мікросервісів та адаптація їх параметрів.
6. Оцінка роботи сценарію та його адаптація для конкретних задач аналізу.
7. Аналіз отриманої інформації та результатів запиту.

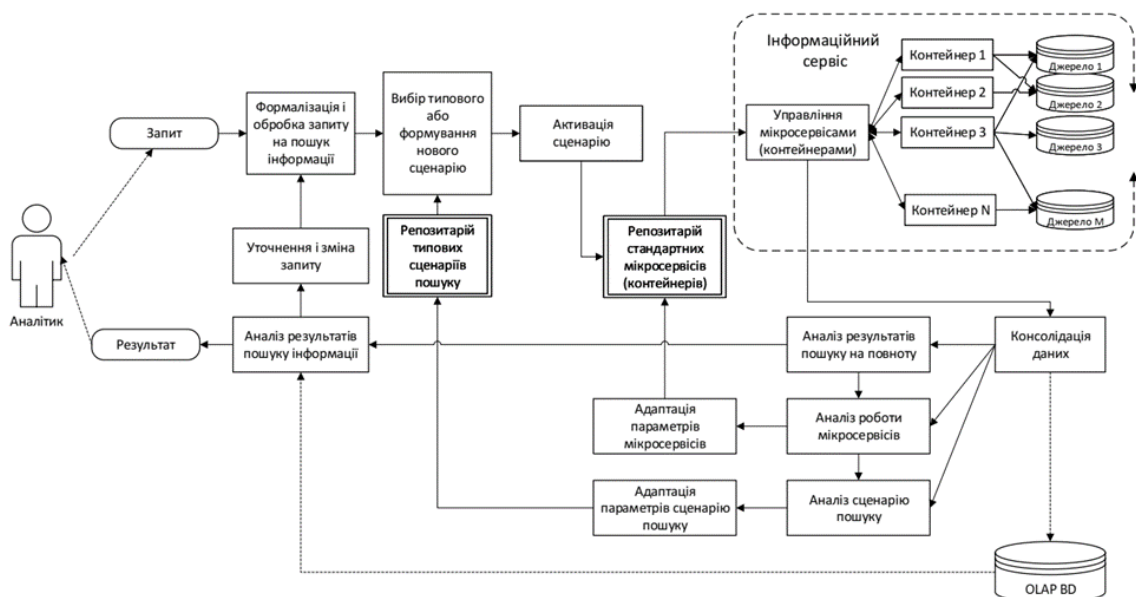


Рис. 18.1. Принципова схема підсистеми збору даних інформації АІС

На першому кроці проводиться аналіз запиту, при необхідності його уточнення та формалізація запиту з точки зору можливості його виконання в рамках

інформаційних та аналітичних можливостей функцій системи. Для уточнюючого та формалізованого запиту обирається типовий сценарій з репозиторію типових сценаріїв, а, при необхідності, формується новий сценарій, як правило, на основі одного з типових сценаріїв.

На другому кроці проводиться підключення до сценою відповідних мікросервісів, як елементів виконання функцій збору та обробки інформації за типовим сценарієм чи його версією. Ця процедура активації сценарію виконується з використанням репозиторію стандартних мікросервісів. При цьому мікросервіси реалізуються як окремі контейнери. Для цього використовують засоби програмної платформи Docker для швидкої розробки, тестування і розгортання мікросервісів. Docker упаковує програмні засоби в стандартизовані блоки, які називаються контейнерами. Кожен контейнер включає все необхідне для роботи програми: бібліотеки, системні інструменти, код і середу виконання. Завдяки Docker можна швидко розгорнути і масштабувати мікросервіси в будь-якому середовищі.

На третьому кроці проводиться виконання збору та обробки інформації за запитом. Для цього використовується платформа Kubernetes, яка дозволяє легко впроваджувати і використовувати додатки мікросервісної архітектури, створюючи рівень абстракції поверх групи хостів [19]. При цьому платформа Kubernetes дає можливість:

1. Розгортання контейнерів і всі операції для запуску необхідної конфігурації. В їх число входять перезапуск зупинених контейнерів, їх переміщення для виділення ресурсів на нові контейнери та інші операції.

2. Масштабування і запуск декількох контейнерів одночасно на великій кількості хостів.

3. Балансування безлічі контейнерів в процесі запуску. Для цього Kubernetes використовує API, завдання якого полягає в логічному групуванні контейнерів. Це дає можливість визначити їх пули, задати їм розміщення, а також рівномірно розподілити навантаження.

На четвертому кроці передбачається консолідація отриманої з різноманітних джерел інформації. Консолідація розглядається як комплекс методів і процедур, спрямованих на об'єднання даних з різних джерел, забезпечення необхідного рівня їх інформативності та якості, перетворення в єдиний формат, в якому вони можуть бути завантажені в сховище даних.

Для збереження отриманої інформації після її консолідації передбачено використання OLAP бази даних, що дає ряд переваг для подальшої обробки в залежності від необхідності аналітика, що надав запит.

П'ятий крок визначає процедури аналізу роботи мікросервісів та адаптацію їх параметрів (рис. 18.2). Адаптація параметрів мікросервісів розглядається як корекція значень параметрів стандартних мікросервісів з репозиторію мікросервісів, що приводить до появи нових версій цих мікросервісів, які будуть закріплені за типовими сценаріями, в результаті виконання яких і з'явилися ті версії.

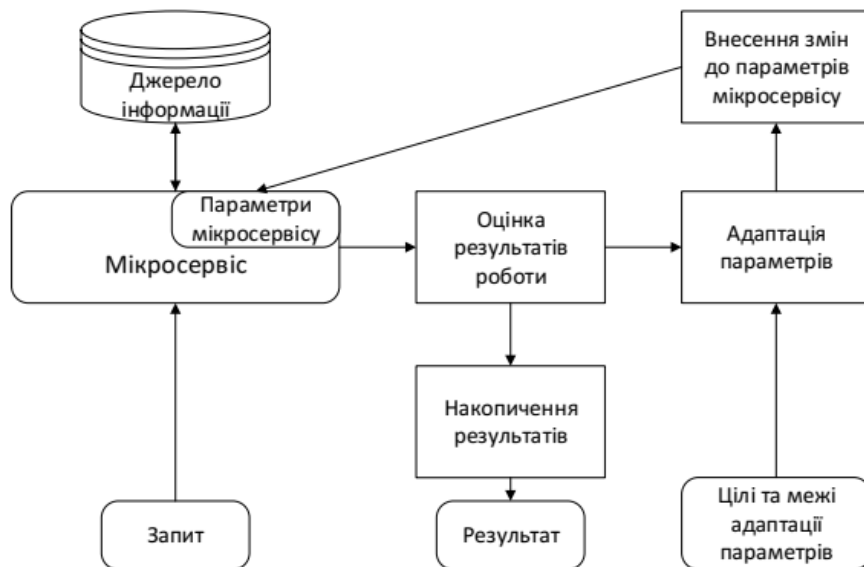


Рис. 18.2. Адаптація параметрів мікросервісу

$A_j = \{a_{1j}, \dots, a_{mj}\}$  – параметри  $j$ -го автомату  $j = \{1, \dots, n\}$ .

При обмеженнях  $a_{ij}^{\min} \leq a_{ij} \leq a_{ij}^{\max}$  для  $i = \{1, \dots, m\}$  та  $j = \{1, \dots, n\}$ .

У загальному випадку обмеження можуть бути як у вигляді меж так і функціональні.

Функція (18.1), що визначає якість роботи  $j$ -го автомату для підсистеми збору інформації, може бути задана як

$$Q_j(A_j) = \frac{k_{ij}}{k_{zj}} \quad (18.1)$$

де

$k_{ij}$  – вдалі спроби пошуку релевантної інформації  $j$ -го автомату,

$k_{zj}$  – загальна кількість спроб  $j$ -го автомату.

Тоді  $\sum_{j=1}^n k_{zj} = M = \text{const}$  – загальна кількість спроб для усіх автоматів на поточній ітерації.

В залежності від результатів роботи мікросервісу на поточні ітерації визначається кількість спроб (18.2), що має виконати мікросервіс на наступній ітерації

$$k_{zj}^{\text{new}} = k_{zj} + \otimes k_{zj} \quad (18.2)$$

при  $\otimes k_{zj} = \Pi(k_{zj}, Q_j(A_j))$ .

У простішому випадку для підсистеми збору інформації значення  $\otimes k_{zj}$  може бути визначено з урахуванням середнього значення роботи усіх автоматів на поточній ітерації.

Наприклад визначення значення  $\otimes k_{zj}$  може бути побудовано на основі порівняння  $Q_j(A_j)$  (18.3) з середнім значенням по всім мікросервісам по поточній ітерації пошуку

$$Q_j^*(A_j) = \frac{\sum_{j=1}^n Q_j(A_j)}{n} \quad (18.3).$$

Тоді якщо

$$Q_j(A_j) \leq Q_j^*(A_j) \text{ тоді } \otimes k_{zj} \leq 0, \text{ а коли}$$

$$Q_j(A_j) > Q_j^*(A_j) \text{ тоді } \otimes k_{zj} > 0.$$

Експериментальні дані показали, що досить ефективним є зменшення чи збільшення значень  $k_{zj}$  в залежності від значень  $Q_j(A_j)$  та  $Q_j^*(A_j)$  на 15-20% з подальшим нормуванням до сумарного значення  $M$ .

Шостим кроком є оцінка роботи сценарію та його адаптація до конкретних задач аналізу рівня міжнародної діяльності. Для побудови нових версій і примірників типових сценаріїв пропонується використання алгоритму побудованого на основі лінійного стохастичного автомату [20]. Структурна адаптація, на основі використання лінійних стохастичних автоматів, базується на оцінці найбільш ефективних джерел інформації, для конкретного запиту та методів обробки інформації, можливих для цих джерел [21]. Це дозволить правильно, по відношенню до найбільшої відповідності, вибрати джерела інформації, відповідно до запиту, та методи обробки цієї інформації.

Для системи оцінки рівня міжнародної діяльності у структурі сценаріїв найбільш критичним є альтернативні методи вилучення інформації, що напряму пов'язані з відповідними джерелами інформації.

Для першої групи важливим параметром якості роботи, що може впливати на адаптацію сценарію, є кількість отриманих релевантних результатів за певний час.

С точки зору пошуку інформації можна визначити:

$u(t)$  – стан системи на поточний момент часу, що визначає ймовірності вибору альтернативних мікросервісів вилучення інформації чи альтернативних мікросервісів обробки інформації ( $M_1, \dots, M_n$ ) на поточній ітерації.

$u(t+1)$  – стан системи на наступний момент часу (на наступній ітерації), що визначає ймовірності вибору альтернативних мікросервісів ( $M_1, \dots, M_n$ ) на наступній ітерації.

Реалізація використання такого автомату для вибору альтернативного мікросервісу може бути побудовано таким чином, що зміна стану автомату  $u(t+1)$  визначається у вигляді регулярної залежності, а його вихід  $u(t)$  визначається у вигляді стохастичного процесу.

Алгоритм, що побудовано на основі використання стохастичного автомату складається з наступних кроків:

Початковий стан (18.4) стохастичного автомату  $u(t)$  для  $t=1$  визначається у вигляді вектору

$$P(t) = \{p_1(t), p_2(t), p_3(t), \dots, p_i(t), \dots, p_{n-1}(t), p_n(t)\} \quad (18.4)$$

де  $p_i(t)$  – ймовірність використання  $i$ -го мікросервісу (18.5) з групи, в якій є  $n$  альтернативних мікросервісів.

$$\sum_{i=1}^n p_i(t) = 1 \quad (18.5)$$

Генерується випадкова величина  $w$ , що рівномірно розподілена на інтервалі  $[0,1]$ .

Випадкова величина  $w$  визначає інтервал, що відповідає одному з  $n$  альтернативних мікросервісів інформації.

Визначаємо (18.6), що  $p_0 = 0$ , тоді якщо

$$\sum_{i=0}^{z-1} p_i(t) < w < \sum_{i=1}^z p_i(t) \quad (18.6)$$

то реалізація автомату буде відповідати мікросервісу з номером  $z$ .

Виконується вибірка чи обробка  $V$  одиниць інформації (документів) мікросервісом за номером  $z$ .

Проводиться оцінка  $R_i$  якості роботи мікросервісу за номером  $z$  для  $V$  одиниць інформації.

За певним для кожного мікросервісу алгоритмом проводиться перерахунок ймовірностей (18.7).

$$k_R = f(R_i) \quad (18.7)$$

Наприклад, коефіцієнт перерахунку ймовірностей стохастичного автомату можна зробити, як показано у таблиці 18.1. Значення  $R_i$  та  $k_R$ , що використані у таблиці, отримані на основі тестування алгоритму і добре показали себе при тестуванні.

Таблиця 18.1 Коефіцієнти ймовірностей стохастичного автомату

$R_i$	0- 0.2<	0.2- 0.4<	0.4- 0.6<	0.6- 0.8<	0.8- 1.0
$k_R$	0.5	1.0	1.5	1.7	2

Тоді 
$$p_z(t+1) = p_z(t)k_R \quad (18.8)$$

Проводиться розрахунок для вектору  $P(t+1)$ .

При цьому (18.9, 18.10, 18.11)

$$D(t) = 1 - p_z(t) \quad (18.9)$$

$$D(t+1) = 1 - p_z(t+1) \quad (18.10)$$

$$p_i(t+1) = \frac{p_i D(t+1)}{D(t)} \quad \text{для } i = 1, \dots, n \text{ та } i \neq z \quad (18.11)$$

Переходимо до кроку 2 для подальшої роботи сценарію з обраним альтернативним мікросервісом, який пов'язаний з певним джерелом інформації чи методом обробки вилученої інформації.

Сьомий, останній крок, відноситься до аналіз отриманої інформації відповідно до запиту. В залежності від оцінки якості отриманої інформації та її відповідності запиту аналітиком, у випадку його невдоволення, проводиться уточнення чи зміна запиту, а в протилежному випадку процес обробки запиту вважається закінченим.

Представлена адаптивна підсистема збору інформації для обробки запитів, що може бути використана у рамках створення інформаційно-аналітичної, що будується для збору та первинного аналізу великих за об'ємом даних, може бути використана при розробці різноманітних інформаційно-аналітичних систем, які орієнтовані на збір інформації на розгалуженій мережі у реальному часі.

## Питання до розділу 18

1. Як розглядається сценарій збору даних в інформаційно-аналітичній системі?
2. Що розуміється під типовим сценарієм збору та обробки інформації?
3. Чим відрізняється мікросервісна мікросервісна архітектура програмних засобів в реалізації сценаріїв від монолітної?
4. Які кроки має процес роботи інформаційно-аналітичної системи?
5. Структуру програмної реалізації підсистеми збору даних?
6. Яка мікросервісна платформа може бути використана при реалізації сценарію збору даних?
7. Як може реалізовуватись адаптація сценарію в інформаційно-аналітичній системі?

## ПЕРЕЛІК ПОСИЛАНЬ

1. Базові методології моделювання бізнес-процесів. URL: [https://stud.com.ua/87179/ekonomika/bazovi\\_metodologiyi\\_modelyuvannya\\_biznes\\_protseviv](https://stud.com.ua/87179/ekonomika/bazovi_metodologiyi_modelyuvannya_biznes_protseviv) (дата звернення 22.10.2023).
2. Георгієв В. А. Економічний сценарій як інструмент управління підприємством / В. А. Георгієв // Менеджмент і маркетинг. – 2016. – № 1 (34). – С. 89–95.
3. Данченко О.Б. Практичні аспекти реінжинірингу бізнес-процесів. К.: Університет економіки та права «КРОК». 2017. 238 с.
4. Долганова О. В. Моделювання бізнес-процесів. 2017.
5. Крижановський Є.М., Яшолт А.Р., Жуков С.О., Козачко О.М. Моделювання бізнес-процесів та управління ІТ-проектами. Вінниця: ВНТУ, 2018. (PDF, 91 с.) URL: [https://ecopy.posibnyky.vntu.edu.ua/txt/2018/Kryzanovsk\\_yascholt\\_modelyuvanna\\_np\\_p024.pdf](https://ecopy.posibnyky.vntu.edu.ua/txt/2018/Kryzanovsk_yascholt_modelyuvanna_np_p024.pdf) (дата звернення 22.10.2023).
6. Томашевський О. М., Цегелик Г. Г., Вітер М. Б., Дудук В. І. Інформаційні технології та моделювання бізнес-процесів. Навч. посіб. – К.: «Видавництво «Центр учбової літератури», 2012. – 296 с.
7. Моделювання бізнес-процесів. URL: [https://stud.com.ua/87161/ekonomika/modelyuvannya\\_biznes-protseviv](https://stud.com.ua/87161/ekonomika/modelyuvannya_biznes-protseviv) (дата звернення 22.10.2023).
8. Пістунов І.М. Моделювання бізнес процесів [Електронне видання]: навчальний посібник / І.М. Пістунов. Електрон. текст. дані. – Д.: НТУ «ДП», 2021. – 130 с. – Режим доступу: [http://pistunovi.inf.ua/MOD\\_BIZ\\_IPOU.pdf](http://pistunovi.inf.ua/MOD_BIZ_IPOU.pdf) (дата звернення: 22.10.2023).
9. Попов С.А. Сценарне моделювання: методика восьми кроків [Електронний ресурс] / С. А. Попов. – Режим доступу: <http://surl.li/odder> (дата звернення: 22.10.2023).
10. BABOK Guide: Global Standard for Business Analysis – ІБА, International Institute of Business Analysis. 2015. URL: <https://www.iiba.org/standards-and-resources/babok/> (дата звернення: 22.10.2023).
11. Godet Michel, Creating Futures: Scenario Planning as a Strategic Management Tool, Economica, 2012.
12. James A. Ogilvy, Creating Better Futures: Scenario Planning as a Tool for a Better Tomorrow, Oxford University Press, 2013.

13. Jay Ogilvy, Peter Schwartz, Plotting Your Scenarios. Global Business Network. December, 2011 – 275p.
14. Kees van der Heijen, Scenarios, Strategies and the Strategy Process. Nijenrode University Press, 2012.
15. Kees van der Heijen. Scenarios – The Art of Strategic Conversation, John Wiley&Sons Limited, 2013 - 187p.
16. Liam Fahey, Robert Randall - Learning from the Future, John Wiley&Sons Limited, 2012 – 114p.
17. Mercer D. Scenarios made easy, Long Range Planning. — Vol. 28. — № 4. 2014. – 283p.
18. Popov S.A. Scenario modeling: a method of eight steps. Retrieved from <http://surl.li/odder> (дата звернення: 22.10.2023).
19. Chris Richardson, Microservices Patterns: With examples in Java, Manning Publications, 2018. — 520 p.
20. Растрингін Л. А. Автоматная теория случайного поиска / Л. А. Растрингін, К. К. Рипа. – Рига: Зинатне (ЛатАН), 1973. – 344 с.
21. Kuzminykh V. O., Otrokh S. I., Shvets Ye.Yu. Analysis of information in the tasks of consolidating from open sources/ V. O. Kuzminykh, // Телекомунікаційні та інформаційні технології. ISSN 2412-4338 – 2018. №1(58) – С. 70-76.
22. UML - Краткое руководство URL: <https://coderlessons.com/tutorials/akademicheskii/uchit-uml/uml-kratkoe-rukovodstvo> (дата звернення 22.10.2023).

## СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Балджи М.Д. Застосування сценарного підходу для проведення прогнозних досліджень на підприємствах / М.Д. Балджи // Науковий вісник Ужгородського університету : зб. наук. праць. Серія «Економіка». 2015. Вип. 2(46). С. 161–165.
2. Данченко О.Б. Практичні аспекти реінжинірингу бізнес-процесів / О.Б. Данченко. – К.: Університет економіки та права «КРОК», 2017. – 238 с. – URL: <http://www.dut.edu.ua/ua/lib/1/category/738/view/566> (дата звернення: 22.10.2023).
3. Коваль З. О. Застосування методу сценаріїв у дослідженні стратегій підприємств / З. О. Коваль // Обліково-аналітичне забезпечення системи менеджменту підприємства : матеріали IV Міжнародної науковопрактичної конференції, 24-26 жовтня 2019 року, Львів. — Львів : Видавництво Львівської політехніки, 2019. — С. 98.
4. Коцко Т.А., Шеховцова І.А. Реінжиніринг бізнес-процесів як технологія антикризового управління підприємствами паливно-енергетичного комплексу / Економіка та суспільство. – №9. – 2017 р. – С. 466-472. – Режим доступу: [http://www.economyandsociety.in.ua/journal/9\\_ukr/80.pdf](http://www.economyandsociety.in.ua/journal/9_ukr/80.pdf) (дата звернення: 22.10.2023).
5. Коюда В.О. Бізнес-процеси сучасного промислового підприємства / В.О. Коюда, М.І. Пасько // Бізнес Інформ. – 2018. – №1. – С. 302-311. – Режим доступу: [http://nbuv.gov.ua/UJRN/binf\\_2018\\_1\\_45](http://nbuv.gov.ua/UJRN/binf_2018_1_45) (дата звернення: 22.10.2023).
6. Лапшин В. С. Управление процессами: учеб. пособие /. – Саранск : Изд-во Мордов. ун-та, 2015. – 385 с. – URL: <http://www.dut.edu.ua/ua/lib/1/category/738/view/1443> (дата звернення: 22.10.2023).
7. Орловський, Д. Л. Бізнес-процеси підприємства: моделювання, аналіз, удосконалення: навч. посіб. у 2 ч. / Д. Л. Орловський ; Харків : НТУ "ХПІ", 2018. Бізнес-процеси: аналіз, управління, удосконалення. 2018. 432 с.
8. Ставерська Т. О. Методичний підхід до планування на основі сценаріїв / Т. О. Ставерська // Механізми управління економікою України. – Донецьк, 2015. – Вип. 176. – Т.11. – С. 306–315.
9. Сервис имитационного моделирования бизнес-процессов. – Режим доступа: <https://www.bpsimulator.com> (дата звернення 22.10.2023).
10. Шандова Н.В. (2017), «Сценарний підхід до визначення напрямків розвитку підприємства», Вісник ЖДТУ 2017. № 1 (79) – режим доступу: <http://ven.ztu.edu.ua/article/view/96700/92371> (дата звернення 22.10.2023).
11. Шуляр Р.В. Розвиток економіко-управлінського інструментарію забезпечення бізнес-процесів: моделювання, регулювання та економічне

обґрунтування / Монографія. Львів: Видавництво Львівської політехніки, – 2018. – 276 с.

12. Cairns, G. Augmented Critical Scenario Method: Analysing stakeholders' achievement of their objectives within future scenarios / G. Cairns, G. Wright // ANZAM. – 2012. – URL: <https://core.ac.uk/download/pdf/78105391.pdf> (дата звернення: 22.10.2023).

13. Culot, G. The future of manufacturing: A Delphi-based scenario analysis on Industry 4.0. / G. Culot, G. Orzesb, M. Sartora, G. Nassimbenia. // Technological Forecasting & Social Change 157. – 2020. – 120092. – Pp. 1-34.

14. Giaoutzi, M. Scenario analysis in foresight: AG2020 Research Memorandum / M.Giaoutzi, A.Stratigea, E.S. Van Leeuwen, P.Nijkamp. – 2011- URL: [https://www.researchgate.net/publication/265064439\\_Scenario\\_analysis\\_in\\_foresight\\_AG2020\\_Research\\_Memorandum\\_2011-37](https://www.researchgate.net/publication/265064439_Scenario_analysis_in_foresight_AG2020_Research_Memorandum_2011-37) (дата звернення: 22.10.2023).

15. Nishadha (2021). Business Process Modeling: Definition, Benefits and Techniques. URL: <https://creately.com/> (Last accessed: 22.10.2023).

16. Nguyen, M.-T. Some Methods for Scenario Analysis in Defence Strategic Planning / M.-T. Nguyen & M. Dunn. – Published by Joint Operations Division DSTO Defence Science and Technology Organisation Fairbairn Business Park, Department of Defence, Canberra, ACT 2600. – 49 p. – URL: <https://indianstrategicknowledgeonline.com/web/ADA498161.pdf> (дата звернення: 22.10.2023).

17. Pinto, T. Strategic bidding in electricity markets: An agent-based simulator with game theory for scenario analysis / T. Pinto, I. Praca, H. Morais, T.M. Sousa. // Integrated Computer-Aided Engineering. – 2013. – 20(4). – Pp. 335-346

18. Quiceno, G. Scenario analysis for strategy design: A case study of the Colombian electricity industry / G. Quiceno, C. Álvarez, R. Ávila, Ó. Fernández, C.J. Franco, M. Kunc, I. Dyner. // Energy Strategy Reviews. – 2019. – 23. – Pp. 57-68.

19. Riaz, S. Computationally Efficient Market Simulation Tool for Future Grid Scenario Analysis / S. Riaz, G. Verbič, A.C. Chapman. // IEEE Trans. Smart Grid. – 2019. – vol. 10. – no. 2. – Pp. 1405-1416

20. Shantiko, B., Liswanti. N. Participatory Prospective Analysis (PPA): Scenario building approach for improving, exploring and anticipating challenges in complex multi-stakeholders systems. – URL: [https://www1.cifor.org/fileadmin/subsites/colupsia/documents/IASC\\_Scenarios.pdf](https://www1.cifor.org/fileadmin/subsites/colupsia/documents/IASC_Scenarios.pdf) (дата звернення: 22.10.2023).

21. Swart, R.J. The problem of the future: sustainability science and scenario analysis / R.J. Swart, P. Raskin, J. Robinson. // Global Environmental Change. – 2014. – 14. – Pp. 137-146