

**МАГІСТЕРСЬКА РОБОТА**

**МР. ШМ - 11.00.00.000 ПЗ**

**Група ШМ-22-1**

**Мальована Діана**

**2024**

**Івано-Франківський національний технічний університет нафти і газу**

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

**Мальована Діана Валентинівна**

(прізвище, ім'я, по батькові)

УДК 004.942  
(індекс)

## **МАГІСТЕРСЬКА РОБОТА**

**Моделі, методи та алгоритми процесів підтримки прийняття рішень**

**основі технологій OLAP і Data Mining**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Мальована Д.В.**

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник

**Корнута Володимир Андрійович, к.т.н., доцент**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Допущено до захисту**

В.о. завідувача кафедри

доц.

**Бандура В.В.**

(посада)

(підпис)

(дата)

(ініціали та прізвище)

Рецензент

доц.

(посада)

(підпис)

(дата)

(ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

**Івано-Франківський національний технічний університет нафти і газу**

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

В.о. зав. кафедрою ШЗ

доц. В.В. Бандура

“ 04 вересня 2023 р.

# ЗАВДАННЯ

## НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Мальованій Діані Валентинівні**

(прізвище, ім'я, по-батькові)

**1. Тема магістерської роботи “ Моделі, методи та алгоритми процесів підтримки прийняття рішень основі технологій OLAP і Data Mining”**

керівник проекту (роботи) Корнута Володимир Андрійович, к.т.н., доцент

затвержені наказом закладу вищої освіти від “ 18 ” грудня 2023 р. № 738/7

**2. Строк подання студентом проекту (роботи) 15 січня 2024 р.**

**3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій СППР**

**4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Дослідження систем підтримки прийняття рішень

2. Дослідження побудови та проектування архітектури систем підтримки прийняття рішень

3. Моделі та методи обробки даних в системах підтримки прийняття рішень

4. Реалізація інформаційної системи підтримки прийняття рішень засобами Data Mining

**5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

1. Місце систем прийняття рішень в інформаційних системах управління (рис. 1.1)

2. Система прийняття рішень базована на знаннях (рис. 1.3)

3. Класифікація засобів підтримки прийняття рішень (рис. 2.2)

4. Загальна модель зберігання даних (рис. 2.10)

5. Прогнозування реалізації продукції засобами Data Mining (рис. 3.8)

## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Нормоконтроль	доц., к.т.н. Вовк Р.Б.	
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2023 р.

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	01.10.2023	виконано
2	Аналіз концепцій та алгоритмів предметної області	25.10.2023	виконано
3	Дослідження побудови та проектування архітектури систем підтримки прийняття рішень	10.11.2023	виконано
4	Моделі та методи обробки даних в системах підтримки прийняття рішень	22.11.2023	виконано
5	Реалізація інформаційної системи підтримки прийняття рішень засобами Data Mining	01.12.2023	виконано
6	Реалізація функціональності запропонованої інформаційної технології	15.12.2023	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.01.2024	виконано

Студент – магістр

\_\_\_\_\_ (підпис)

Керівник роботи

\_\_\_\_\_ (підпис)

## АНОТАЦІЯ

**Магістерська робота:** 103 с., 30 рис., 4 табл., 47 джерел, 1 додаток.

**Тема:** Моделі, методи та алгоритми процесів підтримки прийняття рішень основі технологій OLAP і Data Mining

**Об'єкт дослідження:** моделі, методи та алгоритми побудови ефективної системи підтримки прийняття рішень.

**Мета роботи:** дослідження методичних засад і розробка практичних підходів щодо підвищення ефективності діяльності підприємства за рахунок інформаційної підтримки прийняття управлінських рішень на основі технологій OLAP та Data Mining.

**Предмет дослідження:** інформаційні процеси та методи формування ефективної системи підтримки прийняття рішень на основі OLAP-технологій та Data Mining.

**Результати дослідження:**

Досліджено моделі, методи та процеси створення інформаційної системи підтримки прийняття рішень на основі OLAP-технологій та Data Mining, яке забезпечує підвищення ефективності планування виготовлення продукції.

**Висновок:**

В результаті досліджень було спроектовано та розроблено додаток для інформаційної підтримки прийняття рішень на основі методів Data Mining в задачах планування виробництва для підприємств.

ІНФОРМАЦІЙНА СИСТЕМА, ОНТОЛОГІЯ, ВИДОБУВАННЯ ДАНИХ, МОДЕЛЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ, СИСТЕМА УПРАВЛІННЯ, OLAP-ТЕХНОЛОГІЯ.

## ANNOTATION

**Master's work:** 103 pp., 30 fig., 5 tab., 47 sources, 1 addition.

**The purpose:** Models, methods and algorithms of decision-making support processes based on OLAP and Data Mining technologies

**Object of study:** models, methods and algorithms for building an effective decision support system.

**Purpose of work:** the study of methodological principles and the development of practical approaches to increase the efficiency of the company's activities due to information support for making management decisions based on OLAP and Data Mining technologies.

**Subject of study:** information processes and methods of forming an effective decision support system based on OLAP technologies and Data Mining.

**Research results:**

The models, methods and processes of creating an information system for supporting decision-making based on OLAP technologies and Data Mining, which ensures an increase in the efficiency of product manufacturing planning, have been studied.

**Conclusion:**

As a result of research, an application was designed and developed for informational support of decision-making based on Data Mining methods in production planning tasks for enterprises.

INFORMATION SYSTEM, ONTOLOGY, DATA MINING, MODEL, SOFTWARE, DECISION SUPPORT, MANAGEMENT SYSTEM, OLAP TECHNOLOGY.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
РОЗДІЛ 1	
ДОСЛІДЖЕННЯ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ .....	13
1.1 Системи прийняття рішень в інформаційних системах та методики обробки знань .....	13
1.2 Дослідження інформаційних систем підтримки прийняття рішень .....	18
1.3 Дослідження систем підтримки прийняття рішень, їх класифікація та загальна архітектура на основі OLAP-технології.....	23
1.4 Висновки до розділу.....	31
РОЗДІЛ 2	
ДОСЛІДЖЕННЯ ЕТАПІВ ПОБУДОВИ ТА ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ.....	32
2.1 Особливості систем підтримки прийняття рішень на основі онтологій ....	32
2.2 Дослідження етапів проектування систем підтримки прийняття рішень та їх класифікація.....	38
2.3 Аналіз моделей і критеріїв для СППР .....	48
2.4 Дослідження підходів до проектування СППР .....	50
2.5 Структура та алгоритм роботи СППР .....	52
2.6 Дослідження типів архітектури спеціалізованих СППР .....	56
2.7 Аналіз моделей та технологій представлення даних у сховищах даних ....	58
2.8 Висновки до розділу.....	65
РОЗДІЛ 3	
МОДЕЛІ ТА МЕТОДИ ОБРОБКИ ДАНИХ ТА РЕАЛІЗАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЗАСОБАМИ OLAP І DATA MINING...	67
3.1 Дослідження та описання алгоритмів, на яких базується система обробки даних та генерування результатів .....	67

	7
3.2 Дослідження аналізу даних технологією Data Mining.....	69
3.3 Реалізація архітектури СППР з використанням OLAP-технологій і Data Mining.....	71
3.4 Формалізація та проектування системи підтримки прийняття рішень .....	77
3.5 Підготовка управлінських рішень на основі аналізу даних OLAP та методами DataMining .....	80
3.6 Опис та програмна реалізація системи підтримки прийняття рішень.....	85
3.7 Висновки до розділу.....	89
ВИСНОВКИ .....	91
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	92
ДОДАТКИ .....	97

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

OLAP	-	Online analytical processing
АСУ	-	Автоматизована система управління
ІС	-	Інформаційні системи
IFPS	-	Interactive Financial Planning System
АНР	-	Analytical Hierarchy Process
DSS	-	Decision Support System
ОПР	-	Особа, що приймає рішення
ТПР	-	Теорія прийняття рішень
БЗ	-	База знань
МС	-	Мовна система
СПР	-	Система представлення результатів
ЕТ	-	Електронна таблиця
СОДГР	-	Система обробки даних та генерування результатів

## ВСТУП

### **Актуальність роботи**

Сучасні інформаційні системи, використовувані на харчових підприємствах України, накопичують великі обсяги господарської інформації про бізнес-діяльність підприємства. Завдяки стандартизації та уніфікації звітних документів, що регламентують діяльність кожного підприємства, бази даних харчових підприємств мають схожу структуру економічної інформації.

Актуальним у процесі прийняття рішень є дослідження фактору фінансового ризику. Ризик в економічній діяльності - це об'єктивний фактор, який зумовлений дією стохастичних причин і чинників, зокрема конфліктністю ситуації прийняття рішень, невизначеністю цілей і наслідків дій, відсутністю повної і об'єктивної інформації щодо процесів, що відбуваються тепер чи виявлять себе в майбутньому. Ризик і невизначеність — постійні джерела небезпеки прийняти несприятливе рішення [1].

Кількісна оцінка пріоритетності альтернативних дій залежить від особистих характеристик керівника і тому повинна бути відносною величиною. У наукових джерелах ця категорія називається по-різному: ступінь вигідності, корисність тощо. Теоретичні результати необхідно використовувати як методологічну базу для підготовки та прийняття фінансових рішень у комп'ютерних системах. Тому доцільним і актуальним є дослідження та застосування функції вигідності для прийняття ризикованих фінансових рішень.

Актуальною є задача розробки і впровадження на підприємствах комп'ютерної системи підтримки прийняття рішень (СППР). Розв'язання цієї задачі потребує, у свою чергу, поглибленого вивчення існуючих і розробки нових методів і принципів створення організаційного, інформаційного та програмного забезпечення, розробки економіко-математичних методів та моделей, пов'язаних з управлінням фінансами на мікрорівні.

## **Порівняння роботи з відомими розв'язаннями проблеми**

Сучасні інформаційні технології на основі багатовимірною представлення даних та їх аналізу, розвинуті завдяки роботам Е. Кодда, Б. Інмона, Р. Хакаторна, Р. Кімбола, Н. Пендса, М. Демареста, Е. Спірлі, надають достатньо засобів для формування управлінських рішень. Дослідженнями в області формування алгоритмів та методів інтелектуального аналізу даних займалися такі вітчизняні та зарубіжні вчені, як В.М. Глушков, О. Г. Івахненко, М. З. Згуровський, Г. П'ятецький-Шапіро, У. Файад, Р. Сміт та інші. Їх досвід широко використовується в банківській сфері для аналізу надійності клієнтів, в торгівлі для аналізу продажів, на промислових підприємствах для аналізу збуту.

Дослідження в напрямку підвищення ефективності планування підприємства, за рахунок інтелектуального аналізу даних, розвинені недостатньо.

### **Мета і задачі дослідження**

**Метою** магістерської роботи є дослідження методичних засад і розробка практичних підходів щодо підвищення ефективності діяльності підприємства за рахунок інформаційної підтримки прийняття управлінських рішень на основі технологій OLAP та Data Mining.

Для досягнення поставленої мети вирішено наступні основні завдання:

1. Досліджено інформаційне забезпечення систем підтримки прийняття рішень.
2. Змодельована структури даних для визначених обмежень цільової функції прибутку у вигляді OLAP-кубів.
3. Побудовано функціональну модель та архітектуру СППР, що описує процес підготовки управлінських рішень на основі інтелектуального аналізу даних.
4. Розроблено алгоритм управління собівартістю продукції на основі використання OLAP-технологій та Data Mining.
5. Здійснено практичну реалізацію СППР для підготовки управлінських рішень на основі представлення даних у вигляді аналітичних конструкцій.

**Об'єктом дослідження** є моделі, методи та алгоритми побудови ефективної системи підтримки прийняття рішень.

**Предметом дослідження** є інформаційні процеси та методи формування ефективної системи підтримки прийняття рішень на основі OLAP-технологій та Data Mining.

### **Методи дослідження**

Теоретичною та методологічною основою дослідження стали основні принципи системного аналізу, функціонального аналізу, теорії багатовимірних моделей даних, теорії баз даних. Використано методи структурного моделювання, теоретичні основи проектування реляційних та багатовимірних баз даних, теоретичні основи побудови сховищ даних, основи інтелектуального аналізу даних.

### **Наукова новизна отриманих результатів**

Полягає у розробці інформаційної технології підготовки управлінських рішень, яка на відміну від існуючих підходів, забезпечує пошук закономірностей у багатовимірних масивах даних методами Data Mining.

### **Практичне значення одержаних результатів**

Одержана реалізація системи підтримки прийняття рішень полягає у рішенні, яке забезпечує підвищення ефективності планування виготовлення продукції на основі використання облікових даних підприємства методами багатовимірного аналізу та засобами Data Mining.

### **Особистий внесок студента**

Основним результатом є:

1. Вдосконалена модель інтелектуального аналізу даних за допомогою OLAP та Data Mining з метою моніторингу, аналізу та планування;

2. Виконана підготовка управлінських рішень на основі аналізу даних OLAP та методами Data Mining;

2. Розроблена архітектура та прототип СППР для вирішення задач ефективного планування виробництва підприємства.

### **Структура та обсяг магістерської роботи**

Магістерська робота викладена на 103 сторінках друкованого тексту, який складається із вступу, трьох розділів, висновків, списку використаних джерел (47 найменувань). Робота містить 4 таблиці, 30 рисунків та 1 додаток.

# РОЗДІЛ 1

## ДОСЛІДЖЕННЯ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕННЯ

### 1.1 Системи прийняття рішень в інформаційних системах та методики обробки знань

Інформаційні системи розвиваються більше ніж півсторіччя з 1960х — 1970х років. Спочатку інформаційні системи будувалися як автоматизовані системи управління підприємством. Ці системи мали відповідати запитам менеджерів. Такі системи продукували велику кількість необхідних і корисних для менеджерів і управління документів. Розвиток автоматизованих систем управління в подальшому привів до того, що з'явилися системи підтримки прийняття рішень [3].

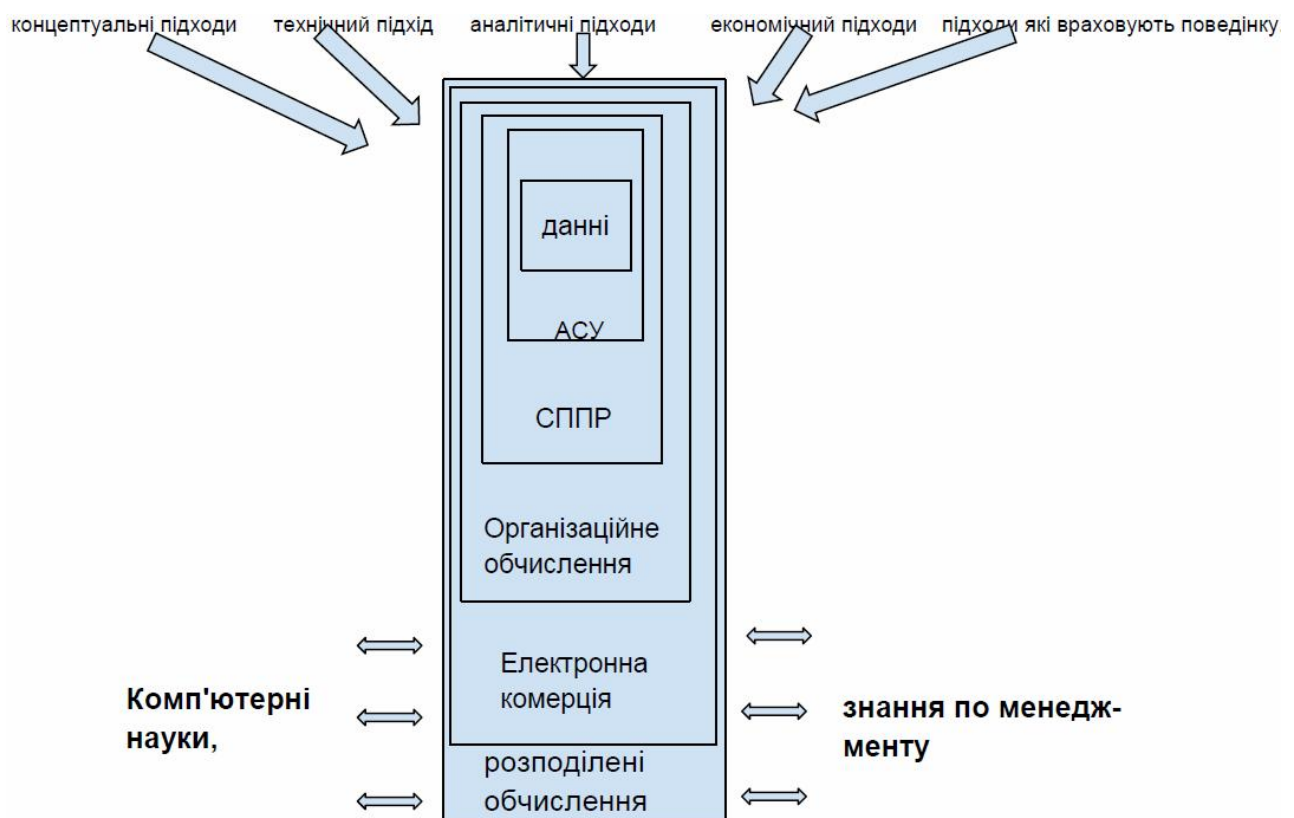


Рис. 1.1. Місце систем прийняття рішень в інформаційних системах управління

Ці системи відрізняються можливостями які дозволяють користувачам отримувати інформацію або знання з інформаційної системи яка була запрограмована і налаштована користувачем так, щоб урахувати досвід користувача і його підприємства, а також можливості математичних моделей та сховищ даних які поліпшують якість прийняття рішень. Подальший розвиток інформаційних систем призводить до побудови обчислювальних середовищ, які включають в себе електронну комерцію, нові можливості аналітики, можливості колективної співпраці, проектування, розробку нової продукції[4].

Взаємозв'язок систем підтримки прийняття рішень з іншими дисциплінами які вивчають і використовують сучасні методи управління показано на рис. 1.1.

Прийняття управлінських рішень можна розглядати як вибір з багатьох можливостей - альтернатив. Наприклад вибір стратегій поведінки, вибір якихось об'єктів. Таким чином, система підтримки прийняття рішень повинна допомагати менеджеру робити такий вибір. Традиційний метод прийняття керівництвом рішень в системах управління показано на рис. 1.2.

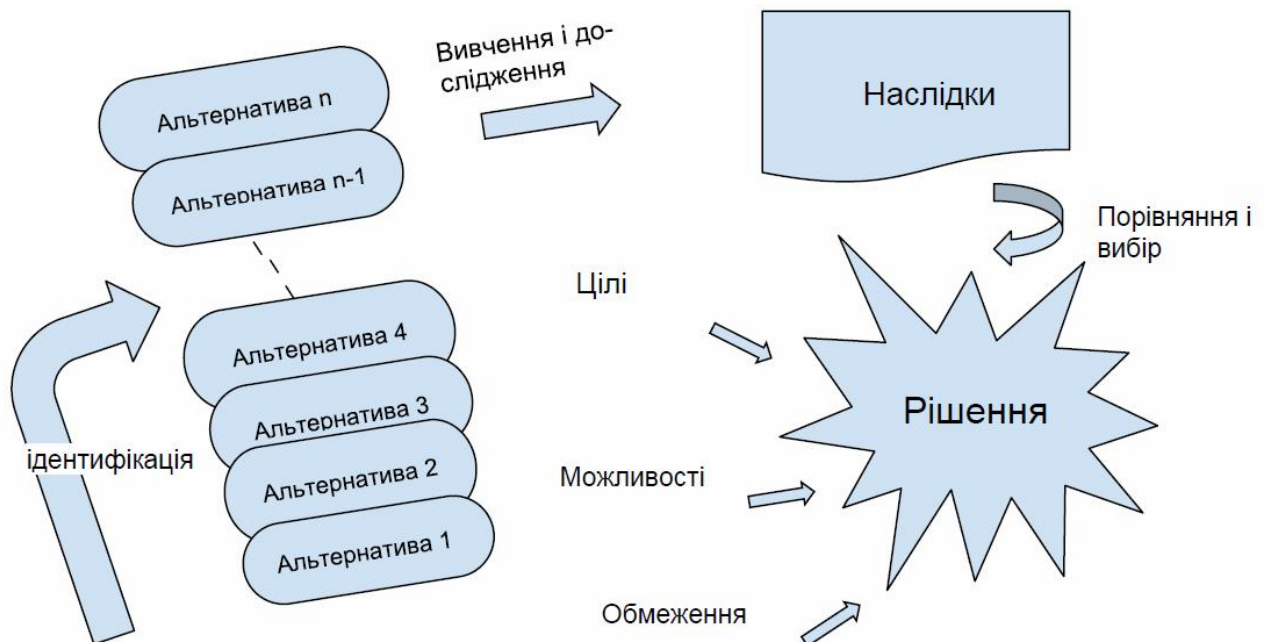


Рис. 1.2. Традиційний метод прийняття рішень

На практиці кількість альтернатив може бути великою, тому менеджер вручну не може ефективно і якісно проаналізувати наслідки прийняття всіх альтернатив. В цій ситуації ефективною є система підтримки прийняття рішень, яка надає можливість за допомогою комп'ютера проаналізувати наслідки всіх альтернатив і змоделювати поведінку системи.

Розвиток науки управління привів до того що традиційний метод прийняття рішень не є найкращим. Розроблено низку нових методів серед яких виділяється метод прийняття рішень базований на знаннях.

Прийняття рішень в системі базованій на знаннях показано на рис. 1.3.

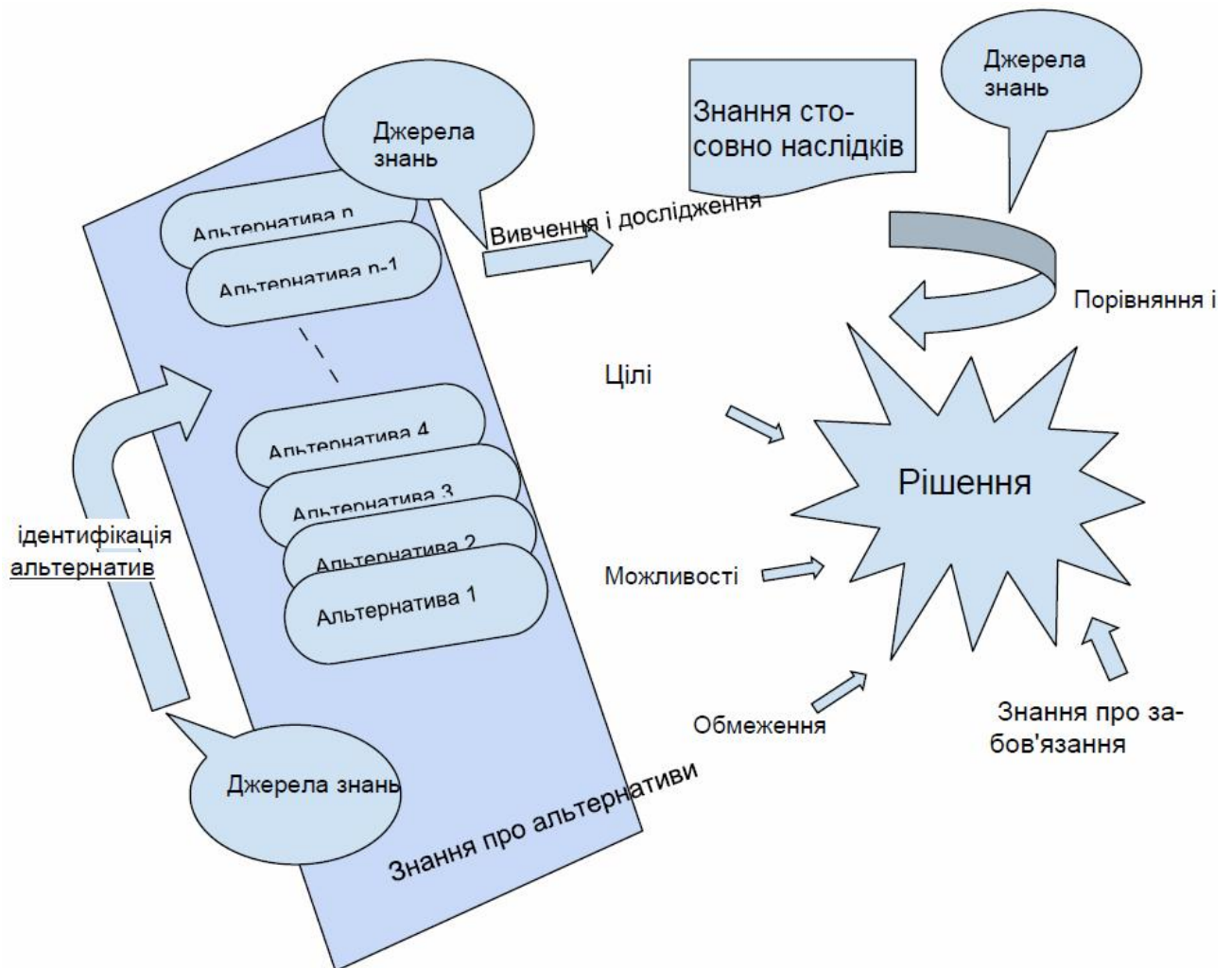


Рис. 1.3. Система прийняття рішень базована на знаннях

В такій системі альтернативи аналізуються не тільки фахівцем, але і за допомогою комп'ютера та програмного забезпечення. В системі досліджуються наслідки застосування альтернатив, далі аналізуються інші джерела знань. Далі

отримані знання обробляються з урахуванням цілей, критеріїв, обмежень і виробляються нові знання які приводять до прийняття науково обґрунтованих рішень. Таким чином система підтримки прийняття рішень являє собою виробничу систему яка виробляє нові знання[5].

Системи підтримки прийняття рішень можуть бути корисними для розробки слабо структурованих рішень. Особливості таких рішень показано в таблиці 1.1. СППР дають можливість виробити правильне рішення в умовах невизначеності чи часткової невизначеності та браку інформації та знань.

Таблиця 1.1

## Системи підтримки прийняття рішень

Структуровані рішення	Неструктуровані рішення
Регулярні, повторювані рішення	Ненадійні, рідко повторювані рішення
Засновані на стабільних умовах	Засновані на випадкових і умовах і даних
Альтернативи чіткі	Альтернативи неясні
Наслідки альтернатив визначені	Наслідки альтернатив невизначені
Критерії вибору чітко визначені	Критерії вибору неоднозначно визначені
Спеціальні знання відомі	Спеціальні знання необхідний для прийняття рішень невідомі
Необхідне знання легко доступні	Необхідне знання недоступно
Заздалегідь відомо повний набір кроків, для досягнення рішення	В процесі прийняття рішень застосовуються не-стандартне мислення, мозковий штурм, синтез, аналогія.
Опора на традиції	Опора на розвідку, творчість, прозоріння, винахідливість

Процес прийняття рішень в СППР полягає у трьох фазах [6].

Перша фаза дослідницька. На цій фазі досліджуються знання та інформація з внутрішніх та зовнішніх джерел. В процесі дослідження розуміння проблеми стає більш ясними і обґрунтованим, альтернативи і їх наслідки більш зрозумілими.

Друга фаза - проектування. На цьому етапі менеджер формулює знання про результати альтернативних дій оцінює їх наслідки для функціонування системи. На цьому етапі менеджер може виробляти додаткові знання.

Третя фаза - вибір рішення. На цьому етапі менеджер виконує вибір між альтернативами з урахуванням їх наслідків які були проаналізовані і досліджені на першій і другій фазах.

Прийняте рішення може бути застосовано не одразу але через деякий час, який підтвердить правильність прийнятого рішення. Всі три фази не завжди йдуть послідовно вони можуть перекривати одна одну і процес прийняття рішення може бути ітеративним.

Метою застосування систем підтримки прийняття рішень є підвищення можливості і здібності менеджера для прийняття правильних і науково обґрунтованих рішень.

Прийняття рішень сучасних системах є наукомістким, тому є необхідними обробка та управління знаннями. Так само, як комп'ютерні науки формують технологічну основу для реалізації систем підтримки прийняття рішень, управління знаннями утворює інтелектуальну основу для розробки, вивчення і застосування систем підтримки прийняття рішень. Алгоритм отримання знань показано на рисунку 1.4.

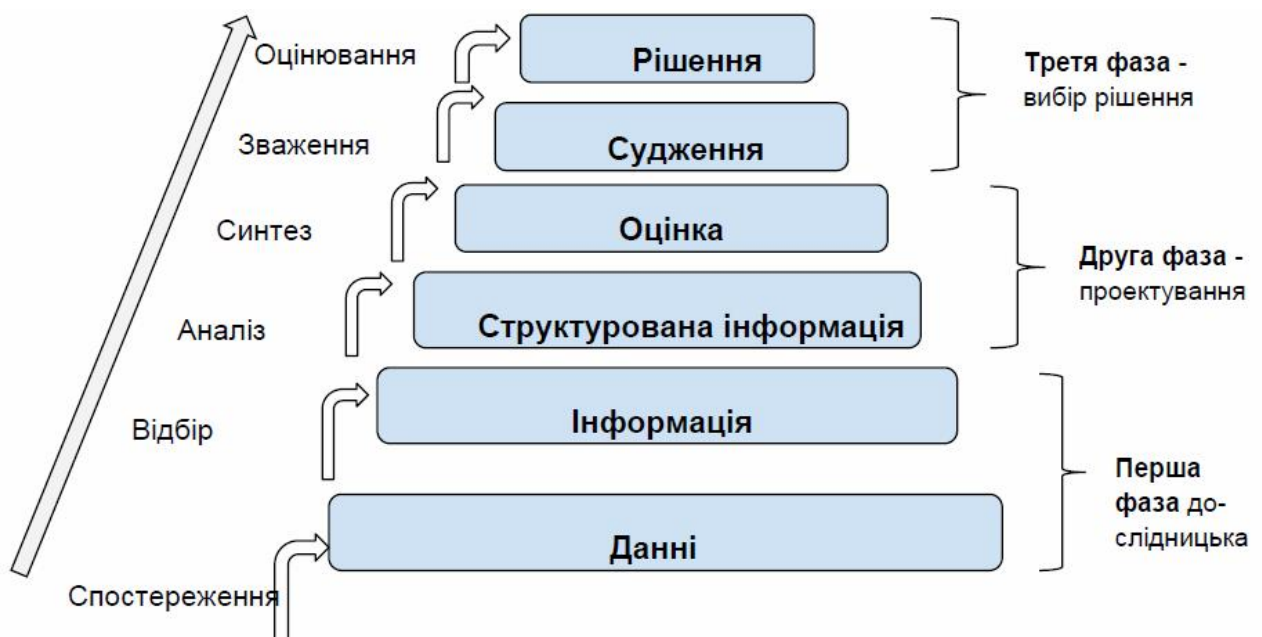


Рис. 1.4. Алгоритм отримання знань

На фазі дослідження менеджер або система управління отримує дані які обробляються і вилучається інформація. Інформація це щось нове, що ми отримали з цих даних. Далі отримана інформація на фазі проектування структурується, приймаються судження які аналізуються і наприкінці приймається управлінське рішення [7].

Розглянемо атрибути знань. Атрибути знань це виміри за якими класифікують знання в СППР. Можливість атрибутів може змінюватися у часі і в залежності від вимог. Тому атрибути які були важливі деякий час назад, можуть стати неважливим в поточному часі. Важливість атрибутів залежить від способів обробки інформації наприклад цифрова чи аналогова. Класифікація атрибутів знань дозволяє створити ефективну систему підтримки прийняття рішень.

Система обробки знань містить щонайменше два основних рівні. Консультаційний центр або люди які надають консультації менеджеру, який приймає рішення утворюють перший рівень обробки знань. Особи які приймають рішення створюють другий рівень обробки знань.

Згідно фундаментальних понять управління знаннями, на першому рівні є п'ять основних класів обробки знань: отримання, відбір, генерація, асиміляція, і розповсюдження.

## **1.2 Дослідження інформаційних систем підтримки прийняття рішень**

Системи підтримки прийняття рішень (СППР) - це особливі інтерактивні ІС, які використовують обладнання, програмне забезпечення, дані, базу моделей і роботу менеджера з метою підтримки всіх стадій прийняття рішень у процесі аналітичного моделювання. Іншими словами, система підтримки прийняття рішень - комплекс програмних засобів, що включає комплекс різних алгоритмів підтримки рішень, базу моделей, базу даних, допоміжні та керівну програми. Керівна програма забезпечує процес прийняття рішень з урахуванням специфіки проблеми [8]. СППР використовується для підтримки різних видів діяльності у процесі прийняття рішень, а саме для:

- полегшення взаємодії між даними, процедурами аналізу й обробки даних і моделями прийняття рішень, з одного боку, й особи, що приймає рішення, як користувача цих систем - з іншого;
- надання допоміжної інформації, особливо для виконання неструктурованих або слабоструктурованих завдань, для яких важко заздалегідь визначити дані та процедури відповідних рішень.

СППР складається з двох основних підсистем - це люди, що приймають рішення, і комп'ютерна система. Якщо менеджеру потрібно скласти виробничий план на рік, то його першим кроком стане створення моделі прийняття рішень за допомогою простої СППР програми.

Систематику СППР можна побудувати за функціональними галузями (маркетинг, планування, інвестиції та ін.), в яких підтримується прийняття рішень, за рівнями інформаційного забезпечення (тактичний, операційний, стратегічний, рівень середньої ланки управління) тощо [9].

Розглянемо дві найвідоміші таксономії СППР.

Класифікація СППР Альтера, розроблена на основі емпіричних досліджень 56 різних СППР, виділяє два типи систем.

1. Системи, орієнтовані на дані (вибирають інформацію):

- накопичування файлів (File drawer systems);
- аналізу даних (Data analysis systems);
- аналізу інформації (Analysis information systems).

2. Системи, орієнтовані на моделі (дають змогу підтримувати прийняття рішень):

- розрахункові або облікові та фінансові моделі;
- репрезентативні або образні;
- оптимізаційні;
- рекомендаційні.

Класифікація СППР Пауера передбачає виділення п'яти категорій СППР (орієнтовані на дані СППР (Data-driven DSS), орієнтовані на моделі СППР (Model-driven DSS), на знання СППР (Knowledge-driven DSS), на документи

СППР (Document-driven DSS), на комунікації та групові СППР і три групи, які ґрунтуються на вторинних ознаках (ін-терорганізаційні та інтраорганізаційні СППР, функціонально-специфічні СППР і СППР загального призначення, СППР на базі Web)[10].

Різноманіття нових інструментів (методи штучного інтелекту, системи інтелектуального аналізу даних, оперативна аналітична обробка - OLAP і технології (World Wide Web, Інтернет, інтернет-мережі) здатне розширити можливості СППР і змінити форми розвитку.

Системи підтримки прийняття рішень набули широкого застосування в економіках різних країн, причому їх кількість постійно зростає. Орієнтовані на операційне управління СППР застосовуються в маркетингу (для прогнозування й аналізу збуту, дослідження ринку і цін), для виконання науково-дослідних і конструкторських робіт, в управлінні кадрами, виробництвом тощо. Найбільша частка комп'ютерної підтримки різних функцій припадає на стратегічне планування, управління і розвиток підприємств, операційне управління й розподіл ресурсів. Розглянемо найвідоміші комерційні реалізації СППР.

СППР DSS-UTES призначена для пошуку оптимальних рішень у складних проблемних моделях. Її побудовано як відкриту систему, що дає змогу підключати для виконання окремих етапів розв'язання задачі автономні проблемні блоки і моделі користувача. До складу DSS-UTES входять: диспетчер системи; підсистема побудови багатовимірної функції корисності; підсистема оцінювання результатів роботи моделі; підсистема оптимізації керівних параметрів програми користувача.

СППР ISDS може бути використана інноваційними та фінансовими менеджерами у формуванні програм розробки нових виробів і технологій у великих корпораціях. Система підтримує такі функції: попередній відбір пропозицій для інноваційних проектів; порівняльний аналіз нових пропозицій і проектів, які вже розробляються; формування інвестиційних груп із проектів, що пропонуються для розробки.

СППР Marketing Expert забезпечує підтримку прийняття рішень на всіх етапах розробки стратегічного і тактичного планів маркетингу та контролю за

їх реалізацією. Система виконує два основні завдання: аудит маркетингу; планування маркетингу з використанням відомих аналітичних методик (GAP-аналізу, сегментного аналізу, SWOT-аналізу, Portfolio-аналізу та ін.).

СППР Decisiyn Grid - програмна оболонка для автоматизації процесу зіставлення дискретних альтернатив за багатьма критеріями. Інформація для прийняття рішень уводиться в порівняльну таблицю, стовпці якої відповідають альтернативам, а рядки - критеріям їх оцінювання. На перетині рядків і стовпців розміщується оцінка альтернативи за певним критерієм. Система має багато функціональних можливостей, які створюють зручні умови для кращого оцінювання і порівняння альтернатив і, отже, підвищують якість процесу прийняття рішень. Є можливість побудови сценарію дій, графічного перегляду результатів, використання шаблонів.

СППР Visual IFPS/Plus (Interactive Financial Planning System) - інтерактивна система планування фінансів. Система дає змогу розв'язувати досить широкий спектр задач: добір балансових підсумків, розподіл прибутку за статтями доходів, передбачення змін валютних курсів, прогнозування, аналіз ризиків, розробки стратегії збуту продукції, відбір науково-дослідних проектів, стратегічне планування, планування прибутку і бюджету, вибір між стратегіями закупівлі та виготовлення продукції власними силами тощо. СППР Visual IFPS/Plus має широке застосування у фінансах, статистиці й управлінні виробництвом.

СППР Analytica - програмне забезпечення кількісного моделювання. Її можливості включають аналіз сценаріїв, діаграми впливу, багатовимірне моделювання й аналіз ризику. Система забезпечує прозорість і потужність бізнес-моделювання, це інструментальний засіб створення й аналізу кількісних бізнес-моделей Analytica широко використовується для створення й дослідження моделей у різних галузях.

СППР Expert Choice орієнтована на застосування діаграм впливу. Expert Choice базується на одному з відомих методів прийняття рішень АНР - багатокритеріальному ієрархічному підході до підтримки прийняття рішень. Ця СППР забезпечує такі можливості: полегшує визначення й описання мети

(цілей); полегшує ідентифікацію всього рангу альтернативних розв'язків; оцінює ключові співвідношення (компроміси) між цілями й альтернативами; дає змогу отримати рішення, яке повністю зрозуміле та підтримується групою розробників проблеми.

Залежно від даних, з якими працюють СППР, вирізняють два основних типи: EIS і DSS. СППР першого типу одержали назву виконавчих інформаційних систем (ВІС). Вони є кінцевими наборами звітів, побудованими на підставі даних із транзакційної інформаційної системи підприємства, що адекватно відображає в режимі реального часу основні аспекти виробничої і фінансової діяльності. Для ВІС характерні такі основні риси:

- звіти базуються на стандартних для організації запитах;
- звіти подаються у зручному вигляді, що включає, разом з таблицями, ділову графіку, мультимедіа тощо;
- орієнтовані на вертикальний ринок, наприклад фінанси, маркетинг, управління ресурсами.

Залежно від специфіки розв'язуваних задач і використовуваних технологічних засобів процесу створення систем можна виокремити такі СППР: спеціалізовані СППР, СППР-генератори, СППР-інструментарій.

Спеціалізовані (прикладні) СППР призначені для використання кінцевими користувачами. Вони дають можливість індивідуальному ОПР чи колективу ОПР справитись зі специфічною множиною пов'язаних проблем у конкретних ситуаціях.

СППР-генератор - це пакет пов'язаних один з одним програмних засобів (пошуку і видачі даних, моделювання тощо), який дає змогу легко і швидко створювати спеціалізовану СППР. Концептуальна структура СППР-генератора, яка відображає користувацьку позицію, включає п'ять компонент: управління інтерфейсом користувача; управління представленням; управління аналізом; системне управління; управління здобуття даних.

СППР-інструментарій охоплює основну область технології, що використовується для побудови СППР, і відповідає найвищому рівню технологічності. Він надає в розпорядження розробників СППР найбільш

потужні програмні засоби, у тому числі спеціалізовані мови, вдосконалені операційні системи, засоби вводу-виводу інформації.

### **1.3 Дослідження систем підтримки прийняття рішень, їх класифікація та загальна архітектура на основі OLAP-технології**

Система підтримки прийняття рішень (СППР) (Decision Support System – DSS) – інтерактивна комп'ютерна автоматизована система (програмний комплекс), що призначена для допомоги та підтримки різних видів діяльності людини при прийнятті рішень стосовно розв'язання структурованих або неструктурованих проблем. Застосування СППР забезпечує виконання ґрунтовного та об'єктивного аналізу предметної області при прийнятті рішень в складних умовах [14].

Задачі прийняття рішень постійно виникають і розв'язуються в природі, у світі що нас оточує – в біологічних, екологічних, соціальних і економічних системах, різноманітних процесах та явищах, наприклад, у процесах функціонування живих організмів та їх колоній, проявах споживчих уподобань, природних катаклізмах тощо.

Рішенням вважається обґрунтований набір дій з боку особи, що приймає рішення (ОПР), спрямованих на об'єкт чи систему управління, який надає можливість привести даний об'єкт чи систему до бажаного стану або досягнути поставленої мети. Рішення є одним із видів розумової діяльності і проявом волі людини. Характерними ознаками рішення є:

- можливість вибору з набору альтернативних варіантів: за відсутності альтернатив, відсутній і вибір, отже, відсутнє й рішення;
- наявність мети: безцільний вибір не розглядається як рішення;
- необхідність вольового акту ОПР при виборі рішення, тому що вона формує рішення при боротьбі мотивів і думок.

Необхідно зазначити, що важливою постає класифікація самих рішень. За існуючими розробками можна виконати класифікацію рішень, яка наведена у табл. 1.2.

Прийняття рішення – це процес вибору найбільш преференційного рішення з множини допустимих рішень або упорядкування множини рішень. Прийняття рішень можливе на підставі знань про об’єкт управління, процеси, що в ньому відбуваються і можуть відбутися з перебігом часу, а також за наявності множини показників, що характеризують ефективність та якість прийнятого рішення. Тобто необхідні адекватна модель об’єкту і модель прийняття та оцінювання прийнятого рішення. Під моделлю прийняття рішень мається на увазі формальне подання поставленої задачі та процесу прийняття рішень [15].

Таблиця 1.2

## Класифікація видів рішень

Ознака	Вид рішення		
Ступінь структуризації проблеми	Гарно структуроване	Погано структуроване	Не структуроване
Кількість етапів реалізації рішення	Статичне (один етап)		Динамічне
Рівень інформованості про стан проблеми	Умови визначеності	Умови ризику	Умови невизначеності
Кількість ОПР	Одна особа		Багато осіб
Зміст рішення	Стратегічне		Тактичне

Питання про формальну основу вибору, зокрема, про походження критерію оптимальності складає одну з фундаментальних проблем теорії прийняття рішень (ТПР). В ТПР були поставлені та досліджені задачі опису і аналізу типів вибору та таких теоретичних конструкцій, як «корисність», «перевага» та ін. Наукові засади ТПР були закладені в період другої світової війни. Її родоначальниками вважаються Дж. Фон Нейман і О.Моргенштерн, які у 1944р. опублікували книгу з теорії ігор. К середині ХХ ст. оптимізаційний вибір за одним чи декількома критеріями був представлений за бінарними відношеннями переваг. В основі сучасних моделей покладені припущення стосовно того, яким чином здійснює вибір варіантів індивідуум, і яким чином здійснюється вибір колективом.

Будь-який процес прийняття рішення здійснюється в декілька основних етапів.

Етап постановки задачі. Складається з фаз аналізу та діагностики проблеми і визначення цілей рішення. На цьому етапі відбувається виявлення та опис проблемної ситуації, збір релевантної інформації і даних; визначаються цілі рішення, яке має бути прийняте, що дозволяє задати напрям пошуку рішень і видалити ті, котрі не відповідають цілям.

Етап формування рішень. Складається з фаз формулювання обмежень і критеріїв прийняття рішень та визначення альтернатив рішення. На даному етапі відбувається визначення обмежень, що дозволяють відокремити прийнятні варіанти від неприйнятних, та критеріїв, які сприяють вибору кращих з придатних варіантів рішення. Потім здійснюється формування множини допустимих альтернатив, яке полягає у пошуку та розробці альтернативних варіантів рішення.

Етап вибору рішення. Складається з фаз оцінки альтернатив та остаточного вибору рішення. На даному заключному етапі відбувається оцінка варіантів з множини допустимих альтернатив за обраними критеріями та подальший остаточний вибір рішення. Цінність альтернативних варіантів звичайно не однакова, але за умов неявної переваги одного варіанту перед іншим можуть виникати певні складності. Процес прийняття рішення складається з таких кроків[16]:

- визначення цілей, критеріїв оптимальності, критеріїв добору «кандидатів» на отримання ресурсів;
- формування множини допустимих альтернатив;
- вибір методів розв'язання задачі;
- порівняння та упорядкування множини альтернатив за обраними критеріями;
- добір кращих варіантів за критерієм оптимальності та вибір рішення.

Часто в процесі прийняття рішень ОПР припускаються помилок. До найбільш поширених належать такі:

- прийняте так зване одностороннє рішення;

- відсутній системний підхід при прийнятті рішення;
- під час вибору варіантів перевага надана «звичній» альтернативі;
- розглядалися лише позитивні варіанти, можливий ризик не було враховано;
- прийняте рішення було зумовлене емоціями;
- рішення прийнято імпульсивно;
- при прийнятті рішення припустились поспішності;
- при прийнятті рішення керувалися припущеннями, прихованими бажаннями і хибними передумовами, а не достовірною суб'єктивною інформацією;
- невірно витлумачені наявні факти;
- неактуальність рішення: рішення було невірно або невчасно реалізоване.

Необхідно зазначити, що будь-яке рішення, має сенс лише тоді, коли воно ефективне. Виділяють два основних фактори, що впливають на ефективність рішень: фактор якості рішення  $Q$  та фактор прийняття рішення людиною  $A$ . Ефективність рішення  $E$  може бути виражена формулою:  $E = Q \cdot A$ .

За умов, що один із зазначених факторів прямує до мінімуму, ефективність рішення падає. Фактор якості рішення  $Q$  пов'язаний із вибором кращої альтернативи з тих, що зумовлює проблемна ситуація з урахуванням умов прийняття рішень та можливостей виконавців рішення [17].

Підвищення ефективності рішення головним чином слід спрямовувати на покращення фактору якості, а саме на вірний добір обмежень і критеріїв рішення, правильне формування множини допустимих альтернатив та на коректний вибір найкращого для умов задачі варіанту.

Так, ефективність розв'язання задачі розподілу ресурсів характеризує ступінь співвимірності досягнутих цілей із витратами ресурсів на їх досягнення та визначається фактором якості рішення  $Q$ , який обумовлюється доброякісністю і глибиною виконання етапу постановки задачі та вибором методів і моделей для розв'язання задачі.

Суттєвий вплив на розв'язок задач прийняття рішень спричиняють умови та середовище, в яких відбувається прийняття рішень [18]. В сучасній ТПР класифікують такі умови прийняття рішень:

- **Визначеність**

Рішення приймається в умовах визначеності, коли точно відомий результат кожного з альтернативних варіантів вибору. Відносно небагато рішень при управлінні бізнес-процесами приймаються в умовах визначеності. Такі ситуації зустрічаються у випадку прийняття рішень, подібних тим, що зустрічались у минулому.

- **Ризик**

До рішень, що приймаються в умовах ризику, відносяться такі, при формуванні яких результати альтернативних варіантів не є визначеними, але відомі їх імовірності. Сума імовірностей всіх результатів певної альтернативи повинна бути рівною одиниці. Зазначимо, що в умовах визначеності існує лише один результат кожного варіанту. Найбільш бажаний спосіб визначення імовірності – об'єктивність. Імовірність є об'єктивною, коли її можна визначити математичними методами або шляхом статистичного аналізу накопиченого досвіду. Також імовірність буде визначена об'єктивно, якщо надійде достатньо інформації для того, щоб прогноз виявився статистично достовірним.

- **Невизначеність**

Рішення приймається в умовах невизначеності, коли неможливо оцінити імовірність потенційних результатів. Така ситуація зазвичай має місце, коли фактори, що необхідно врахувати, є складними, і стосовно їх неможливо отримати достатньо інформації. Тому імовірність певного наслідку неможливо прогнозувати з достатнім ступенем достовірності. Невизначеність є характерною для багатьох рішень, які приймаються у швидко мінливих обставинах.

Середовище, в якому відбувається прийняття рішення, також є важливим фактором, що впливає на процес прийняття і результат прийняття рішення. Хід часу зумовлює ситуаційні зміни. Якщо зміни значні, то ситуація може

змінитися настільки, що обмеження і критерії прийняття рішення стануть недійсними. Тому рішення належить розробляти, приймати і втілювати в умовах, коли інформація та припущення, на яких ґрунтується рішення, залишаються дійсними і актуальними.

Аналіз умов наявності невизначеностей може виконуватись на абстрактному теоретичному рівні або з точки зору конкретної ситуації, наприклад, з точки зору виявлення можливості побудови математичної моделі чи з точки зору теорії інформації, де невизначеність виступає як характеристика ситуації вибору. Категорія невизначеності характеризується деякими змінними параметрами, які описують різні види невизначеностей: глобальну невизначеність, ситуативну, політичну, соціальну і т. д. При розв'язанні задач прийняття рішень в умовах наявності невизначеностей необхідно встановити рівень аналізу і типи невизначеностей, що розглядаються.

Необхідно зазначити, що в реальних практичних задачах прийняття рішень і системного аналізу часто наявні різноманітні види невизначеностей, які разом складають деякий комплекс невизначеностей, так звану системну невизначеність [19].

Проте необхідно зазначити, що процес прийняття рішень людиною має певні обмеження стосовно можливості аналізу, обробки даних, одержання рішень прогнозованої якості та швидкості прийняття обґрунтованих рішень. Робота ОПР обмежена як відносинами між окремими особами, так і внутрішніми психологічними і фізіологічними причинами. Людина має можливість одночасно оперувати лише обмеженим числом операндів і понять, щонайбільше  $7 \pm 2$ . Крім того, при аналізі і розв'язанні багатокритеріальних задач ОПР досить часто проявляють мінливість, невпевненість, нелогічність, намагання суттєво спростити задачу. У таких випадках можливості обчислювальних машин значно перевищують можливості людини, що призвело до розробки напряму розробки систем та методологій, які мають можливість об'єднати переваги людини і комп'ютера та компенсувати їх недоліки – це людино-машинні системи.

На сьогодні не існує єдиної загальної класифікації СППР. Відзначимо, що чудова для свого часу класифікація Альтера, яка розбивала всі СППР на 7 класів, на даний час дещо застаріла. Розглянемо деякі основні поділи СППР на види за різними характеристиками.

На рівні користувача виділяють такі види СППР:

- активна – може зробити пропозицію, яке рішення варто вибрати;
- пасивна – допомагає у процесі ухвалення рішення, але не може внести пропозицію, яке рішення прийняти;
- кооперативні – дозволяє ЛПР змінювати, поповнювати або поліпшувати рішення, пропоновані системою, посилаючи потім ці зміни в систему для перевірки.

На технічному рівні розрізняють такі СППР:

- СППР рівня підприємства – підключена до великих сховищ даних і обслуговує багатьох менеджерів підприємства;
- персональна настільна СППР – мала система, що обслуговує лише один комп'ютер користувача.
- На концептуальному рівні відрізняють такі типи СППР:
  - керована повідомленнями (Communication-Driven DSS) – підтримує групу користувачів, що працюють над виконанням загальної задачі;
  - керована даними (Data-Driven DSS, Data-oriented DSS) – в основному орієнтується на доступ і маніпуляції з даними;
  - керована документами (Document-Driven DSS) – здійснює пошук і маніпулювання неструктурованою інформацією, заданої в різних форматах;
  - керована знаннями (Knowledge-Driven DSS) – забезпечує рішення задач у виді фактів, правил, процедур;
  - керована моделями (Model-Driven DSS) – забезпечує доступ і маніпуляції з математичними моделями (статистичними, фінансовими, оптимізаційними, імітаційними).

Відзначимо, що деякі OLAP-системи, які дозволяють здійснювати складний аналіз даних, можуть бути віднесені до так званих гібридних СППР,

що забезпечують моделювання, пошук і обробку даних та відповідають властивостям декількох видів СППР.

В залежності від типів даних, з якими ці системи працюють, СППР умовно можна розділити на:

- оперативні;
- стратегічні.

Оперативні СППР призначені для негайного реагування на зміни поточної ситуації в керуванні фінансово-господарськими процесами компанії, об'єднання, галузі чи держави.

Такі СППР називають Виконавчі Інформаційні Системи (Executive Information Systems). За суттю, вони представляють собою кінцеві множини звітів, побудовані на підставі даних із транзакційної інформаційної системи оперативного обліку підприємства. Вони забезпечують адекватне відображення в режимі реального часу основних аспектів виробничої і фінансової діяльності підприємства. Для таких СППР характерні такі риси:

- звіти ґрунтуються на стандартних для організації запитах, кількість яких відносно невелика;
- СППР представляє звіти в максимально зручному виді, що включає поряд з таблицями, ділову графіку, мультимедійні можливості і т. п.;
- СППР зазвичай орієнтовані на конкретну сферу, наприклад, фінанси, маркетинг, керування ресурсами.

Стратегічні СППР орієнтовані на аналіз значних обсягів різномірної інформації, що збираються з різних джерел. Найважливішою метою цих СППР є пошук найбільш раціональних варіантів розвитку бізнесу компанії із урахуванням впливу різних факторів, таких як кон'юнктура цільових для компанії ринків, зміни фінансових ринків і ринків капіталів, зміни у законодавстві і т. ін.

Такі СППР припускають достатньо глибоке перетворення даних, спеціально перетворених таким чином, щоб їх було зручно використовувати у процесі прийняття рішень. Невід'ємним компонентом СППР цього виду є правила прийняття рішень, які на основі агрегованих даних дають можливість

менеджерам компанії обґрунтовувати свої рішення, використовувати фактори стійкого росту бізнесу компанії і знижувати ризики. Стратегічні СППР будуються на принципах багатовимірного представлення та аналізу даних (OLAP).

#### **1.4 Висновки до розділу**

Отже, аналіз і проектування є необхідними умовами успішного розроблення інформаційних систем. Знання і управління знаннями є невід'ємною частиною системи управління та систем підтримки прийняття рішень. Існує п'ять методів обробки інформації першого рівня і чотири методи обробки інформації другого рівня. Система підтримки прийняття рішень, яка реалізує або сприяє сполученню цих дев'яти класів обробки знань може реально покращити продуктивність управлінських рішень.

Розробники систем підтримки прийняття рішень мають враховувати наступні фактори:

- Наукомісткій характер процесу прийняття рішень;
- Процес прийняття рішень орієнтується на знання;
- Застосування багаторівневих схем прийняття рішень;
- Застосування дворівневої системи обробки знань яка приймає активну участь в розробці управлінських рішень;
- Застосування в системах підтримки прийняття рішень і методів управління знаннями в сучасній науці та практиці підвищується.

## РОЗДІЛ 2

### ДОСЛІДЖЕННЯ ЕТАПІВ ПОБУДОВИ ТА ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

#### 2.1 Особливості систем підтримки прийняття рішень на основі онтологій

Система підтримки прийняття рішень – це комп'ютеризована система, яка за допомогою аналізу великої кількості інформації може робити прогнози, з врахуванням яких будуть прийматися рішення. Сучасні системи передбачають обробку певних суджень, які розглядалися на різних етапах прийняття рішень. Ці судження можуть бути описані у вигляді об'єктів деякої предметної області, де кожен такий об'єкт є відображенням сукупності знань, з врахуванням яких і приймається рішення. Звідси постає потреба у забезпеченні обробки взаємопов'язаної інформації та її використання у системах з іншим призначенням для точнішого прогнозування.

Застосування методів інженерії знань, зокрема онтологій у системах підтримки прийняття рішень, є доволі поширеним. Так, часто використовують інтелектуальну систему підтримки прийняття рішень, що застосовує графові моделі подання знань. Ситуація прийняття рішення розглядається як скінченний автомат, і задача вибору рішення подається як задача розпізнавання образів.

Близьким змістовно до проблематики мереж онтологій є завдання інтеграції та встановлення відповідності між онтологіями та їхніми складовими. У цій області досліджень розглядають питання про концепти зі змістом, що перекривається. Наскільки відповідають вони один одному семантично.

Для визначення семантичної подібності концептів застосовують міри семантичної близькості. У випадку побудови онтологій на основі аналізу текстів визначеної предметної області для визначення подібності використовують методи математичної лінгвістики.

Враховуючи вищесказане, перспективним рішенням є подання інформації у системі підтримки прийняття рішень у вигляді онтологічних мереж, де кожна така онтологія містить інформацію у певній предметній області, відношення до інших мереж, а також функції, що дають змогу отримати певні дані в процесі прийняття рішення.

Онтологію можна подати :

$$O = \langle T, R, F, \rangle$$

де  $T$  – це терміни предметної області;  $R$  – відношення між ними;  $F$  – функції інтерпретації. Причому при формуванні системи підтримки прийняття рішень множини  $R$  і  $F$  не можуть бути пустими.

Тобто це є певна непуста множина об'єктів, якій приписують такі властивості:

- об'єкти формують деяку ієрархічну структуру;
- терміни інтерпретуються відповідно до функції інтерпретації;
- визначення термінів є аксіомами.

Така онтологія може бути представлена у вигляді певної інформаційної системи, яка складається з подібних термінів та зв'язків між ними, а результатом такої системи буде онтологічна база знань.

Однак, у процесі прийняття рішень є ще один важливий елемент, – це критерій, за яким відрізняються альтернативи з відповідної множини. Саме критерій має велике значення у прийнятті рішення людиною.

Отож, онтологію проблемної ситуації для системи підтримки прийняття рішень можна представити як:

$$O = \langle T, R, F, R \rangle,$$

де  $R$  – це критерії, які є у кожного елемента онтології та використовуються у задачі прийняття рішення. Таку задачу можна описати як множину альтернатив, в якій у кожного елемента є спеціальні значення, що є критеріями цієї альтернативи. А розв'язком такої задачі є підмножина

якнайкращих альтернатив, розташованих за важливістю значення критерію. Причому критерії може змінюватися на усіх етапах прийняття рішень.

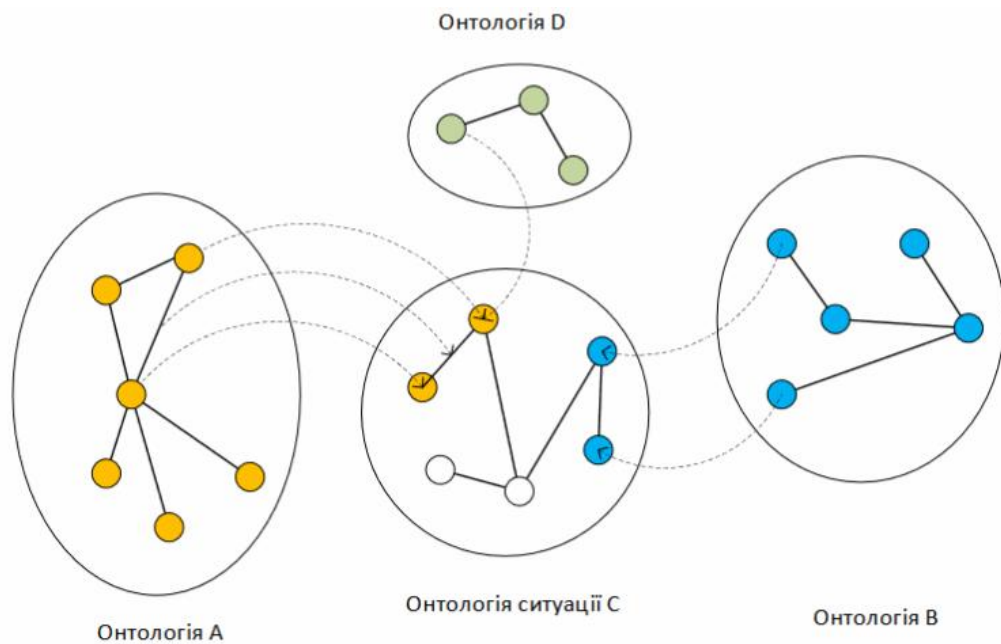


Рис. 2.1. Формування онтології проблемної ситуації у мережі онтологій

Онтологія проблемної ситуації  $O_s$  формується (рис. 2.1) на основі деякого шаблону – моделі ситуації, розроблено експертом. Спочатку він містить елементи та відношення, які після виконання операції встановлення відповідності заміщуються елементами з інших онтологій. При цьому можуть використовуватися різні онтології для відображення різних частин онтології проблемної ситуації. Наприклад, при прийнятті рішення щодо надання студенту відпустки за станом здоров'я можуть знадобитися знання з декількох онтологій: онтології навчального закладу, яка описує деталі діяльності студента у навчальному процесі, особистої онтології, яка визначає майновий та сімейний стан, онтології поліклініки, де визначено перебіг хвороби студента, а також онтології законів про працю. На рис. 2.1 як приклад показано, що онтологія проблемної ситуації  $C$  ініціалізується концептами та відношеннями з онтологій  $A$ ,  $B$  та  $D$ . При цьому один з концептів онтології ситуації сформовано на основі концептів двох онтологій.

На рис. 2.2 подано структуру системи підтримки прийняття рішень з використанням онтологій.

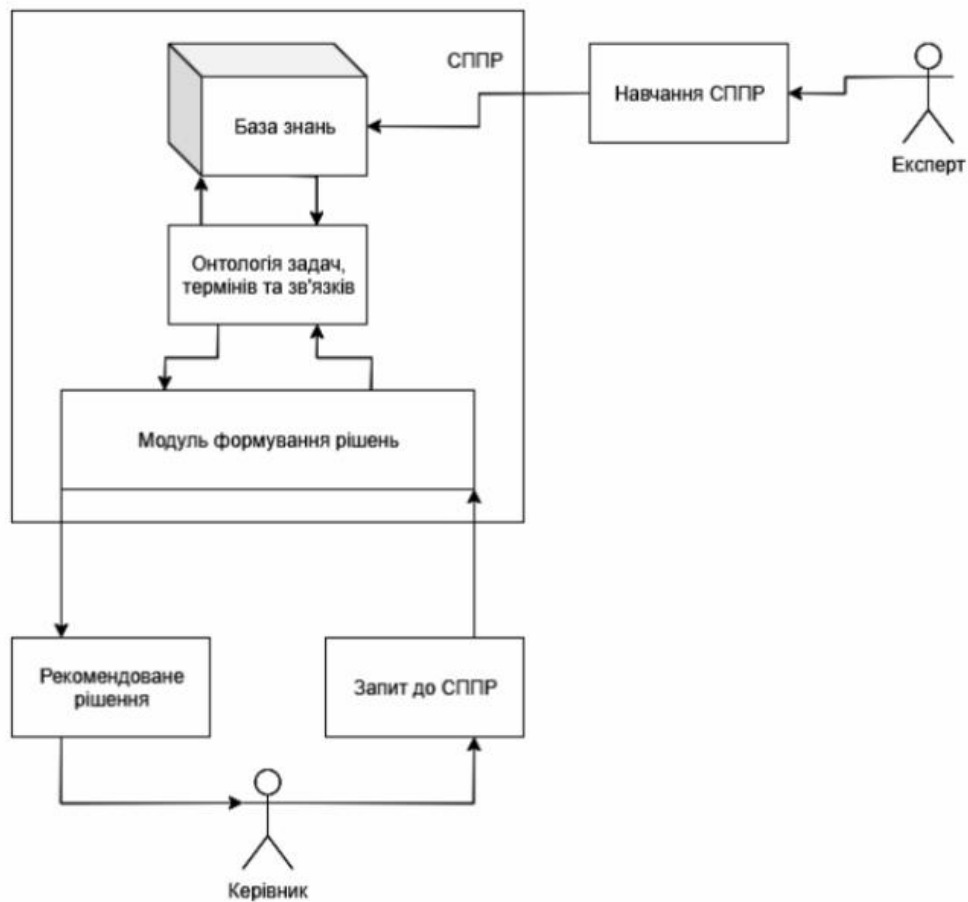


Рис. 2.2. Структура СППР з використанням онтологій

Експерт проводить навчання системи, в процесі якого формується її база знань. Частиною цієї бази знань є онтологія задач, термінів та зв'язків.

Загалом процес прийняття рішення проходить у декілька етапів

1. Постановка задачі.
2. Пошук інформації та її аналіз. Збирається вся доступна інформація на момент прийняття рішення, джерелом якої можуть бути: інтернет, експерти, опитування.
3. Формування критеріїв. На цьому етапі значно допомагають дані, зібрані на попередньому етапі.
4. Побудова онтології, тобто множини альтернатив.

5. Фільтрація альтернатив.

6. Вибір найкращої альтернативи.

Переваги використання онтологій як бази для системи підтримки прийняття рішень:

1. У минулому більшість систем були закритими, сьогодні онтології є відкритими даними, викладеними в інтернеті, що означає: інформація є доступною для всіх.

2. Оскільки дані знаходяться у вільному доступі, створюються системи, які агрегують інформацію з різних веб-сервісів, тим самим створюючи все більші та складніші бази.

Деякі етапи в процесі прийняття рішення можуть бути передані сучасним службам, які надаються компаніями, що в цьому спеціалізуються.

Із врахуванням нових ідей для побудови онтологій з'являється можливість повторного використання існуючих баз знань. Для вирішення завдань підтримки прийняття рішень вже розроблено багато систем. Існують спеціалізовані програми для створення систем підтримки прийняття рішень у таких сферах, як управління продажами чи управління банківськими операціями. Однак, для вирішення проблем у вузьких сферах стає необхідним використання не одного алгоритму, а поєднання кількох.

У такому випадку постає необхідність у створенні базової системи підтримки прийняття рішень, яка працюватиме з онтологічними мережами. Така система виглядатиме як базовий проєкт з готовим пакетом бази знань, яку представляють онтології. На основі такого проєкту під час роботи над інтеграцією системи для вирішення складніших завдань необхідно буде лише доставити посилання на онтології, пов'язані з конкретною задачею.

У такої системи є два входи: один для експерта, задача якого полягає в поповненні/поновленні джерел знань, а другий – для керівника з можливістю задавати запити до системи. Потім йде модуль джерела знань, у якому зберігаються усі підключені онтології та зв'язки між ними. З бази знань формується один із модулів системи підтримки прийняття рішень, а саме база знань, яку використовує ядро.

Модуль ядра відповідає за всі інші задачі, наприклад: прийом запиту від керівника, завантаження необхідних знань із бази знань відповідно до термінів, запуск формування множини альтернатив, вибір альтернатив, що якнайкраще відповідають запиту, вибір найкращої альтернативи та виведення рішення.

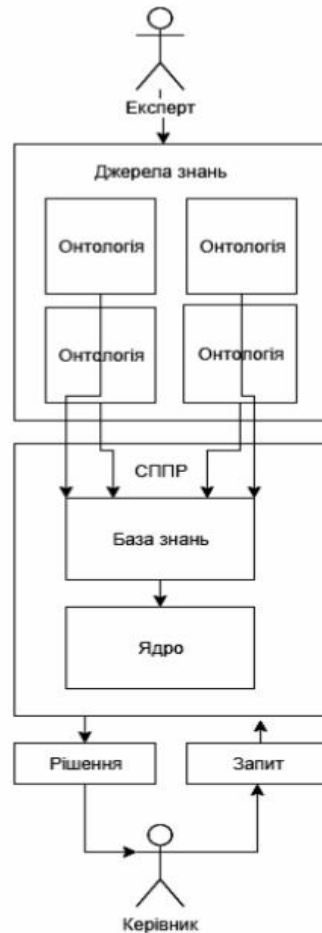


Рис. 2.3. Структура базової СППР із використанням онтологічних мереж

Як видно з рис. 2.3, мережу онтологій використовують на всіх етапах роботи системи. Перш за все це формування бази знань через використання представлених зв'язків, а також формування нових. Та на етапі роботи ядра онтологія допомагає швидко знайти відповідні дані.

Наступною задачею при розробленні базової системи може бути надання можливості редагування бази знань, ядром системи. Це допоможе під час формування рішення при складних запитах, коли рішення, сформоване у попередніх запитах, є важливим для подальших. Рішенням в такому випадку може бути впровадження локальної бази даних між ядром та базою даних.

## 2.2 Дослідження етапів проектування систем підтримки прийняття рішень та їх класифікація

Розглянемо докладніше основні етапи проектування.

**Перший етап.** Визначення і декомпозиція задачі прийняття рішень. Створення опису й аналіз задачі прийняття рішень – це, фактично, основний етап при проектуванні системи. Якщо не встановити із самого початку обмеження на функції і дані системи, то всі подальші зусилля можуть виявитися даремними, оскільки вони можуть виходити за рамки можливостей створюваної СППР. Експерти, що приймають рішення і звикли бачити весь процес «у цілому», можуть мати істотні труднощі з використанням системи, що змусить «звужити» їхнє бачення до досить вузької перспективи. Досвід проектування СППР свідчить про те, що коли СППР «бачить» рішення у вузькому ракурсі, то користувач, що приймає рішення, не може скористатися нею в тім обсязі як планувалося, або ж приймає гірші рішення ніж без системи.

Наведений вище аналіз процесу прийняття рішень людиною створює основу для опису й аналізу задачі прийняття рішень при проектуванні СППР. У процесі виконання своєї задачі проектувальник змушений постійно рухатися у двох напрямках: аналізувати те, що в дійсності має місце в поточній ситуації, і як ОПР сприймає і представляє для себе цю ситуацію. Відповідно до принципу руху до поставленої мети, проектувальник повинен описати ієрархію процесу ухвалення рішення особою, що його приймає. Створювана ієрархія включає перелік подій і проміжних цілей.

Необхідно також об'єднати два напрями мислення – шлях зовнішнього розуміння проблеми і внутрішній когнітивний шлях, по якому йде ОПР. Це дасть можливість уточнити ті сторони процесу прийняття рішень, де необхідно застосувати комп'ютерні засоби для його підтримки.

Декомпозиція задачі виконується в два етапи. На першому етапі встановлюються границі між окремими ситуаціями, що є важливими з погляду ОПР. Ця задача вирішується шляхом декомпозиції головної мети. Потім кожна ситуація докладно описується за допомогою структурованого формату.

Декомпозиція мети. Вище було сказано, що поводження людини має цілеспрямований характер і процес прийняття рішень – це не виключення. Коли приймається рішення, воно має визначену мету. Першим кроком виконання аналізу проблеми, у відношенні якої приймається рішення, є встановлення й опис цілей, що ставить перед собою ЛПР. Декомпозиція цілей виконується у відповідності до двох таких принципів:

- процес прийняття рішень розглядається як прагматичний процес, перед яким поставлена конкретна мета;
- процес прийняття рішень має ієрархічний характер; при цьому спочатку визначаються більш загальні цілі, а потім конкретні проміжні цілі.

Ключовим моментом процесу прийняття рішень, як прагматичного процесу, є висвітлення тих подій, що знаходяться в центрі уваги ОПР. Що відбудеться, якщо було прийняте правильне рішення і досягнута поставлена мета. У більшості випадків ОПР має чітке представлення про процес і шляхом опитування легко встановити, яку конкретну мету вона поставила перед собою. Кожну дію, що повинна мати місце у процесі ухвалення рішення, називають цільовою подією. Як правило, це фізичні події, такі як підвищення доходу до заданого рівня, досягнення бажаних характеристик системи і т.п.

Після того як визначено мету верхнього рівня, необхідно докладно заповнити щонайменше один рівень під кожною метою верхнього рівня. ОПР встановлює які кроки (етапи), дії, функції необхідно виконати щоб домогтися поставленої мети. Їх можна визначити як цілі нижнього рівня, що спрямовані на досягнення мети верхнього рівня. Кожна мета верхнього рівня, що впливає з декомпозиції задачі, має свій конкретний контекст для ухвалення рішення. Для досягнення головної мети верхнього рівня ОПР встановлює компроміс між усіма цілями нижнього рівня і функціями, що необхідні для досягнення головної мети. Діяльність, зв'язана з досягненням деякої мети верхнього рівня, визначається як ситуація ухвалення рішення для цієї мети.

Філософія проектування СППР ґрунтується на тому, що підтримка прийняття рішень повинна зосереджуватися на ситуативному рівні, навколо головної мети, для досягнення якої створюється СППР. Навіть якщо СППР

використовується для підтримки прийняття рішень на нижньому рівні ієрархії управління, все рівно вона повинна бути орієнтованою на ту ситуацію ухвалення рішення, частиною якої є мета нижнього рівня. Це необхідно для того, щоб інтегрувати подальше рішення, прийняте стосовно нижнього рівня, в процес ухвалення рішення щодо головної мети. Таким чином, проектувальник системи повинен побудувати щонайменше один інструмент підтримки прийняття рішень для досягнення кожної мети верхнього рівня [25].

Якщо між цілями самого верхнього рівня існують взаємозв'язки і досягнення цілей повинне виконуватися паралельно, то необхідно проектувати систему із урахуванням цих можливостей і потреб. Необхідно вказати, що СППР не повинна проектуватися для досягнення мети проміжного рівня, навіть якщо це найпростіший шлях з обчислювальної точки зору. Із сказаного випливає, що після виконання декомпозиції головної мети подальший аналіз і опис повинні будуватися таким чином, щоб процес проектування був зосереджений на індивідуальних ситуаціях прийняття рішень.

Формат опису ситуації з ухвалення рішення складається з таких восьми категорій (рис. 2.4):

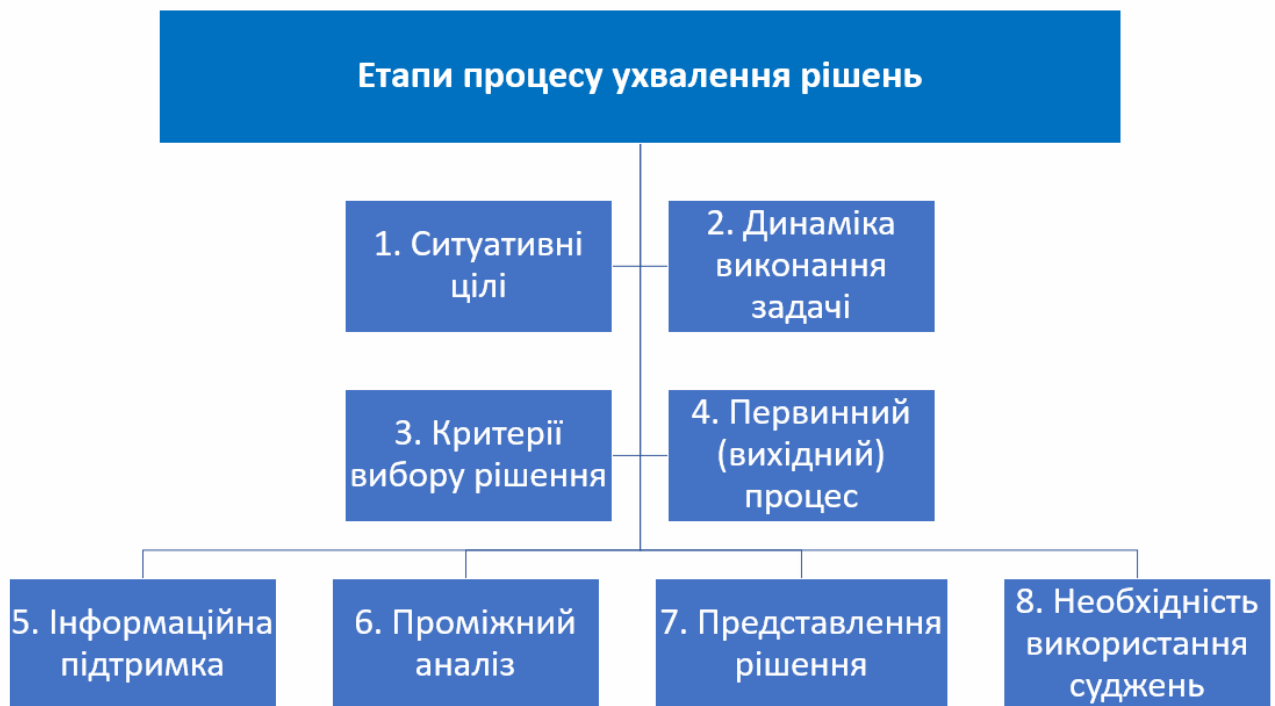


Рис. 2.4. Представлення етапів ухвалення рішень

**1. Ситуативні цілі.** У цій частині документуються результати декомпозиції головної мети. Описується головна мета верхнього рівня, на якій зосереджується процес прийняття рішень і когнітивна обробка даних у конкретній ситуації з прийняття рішень.

**2. Динаміка виконання задачі.** В іншій частині ситуативного опису визначається динамічний контекст, до якого відноситься конкретна ситуація. При цьому можливі такі альтернативи:

- ітеративний аналіз і створення опису – ситуація, що була проаналізована, розглядається знову;
- послідовний аналіз ситуативних даних, коли наступний крок розглядається як наслідок виконання попереднього;
- простий опис, що не має потреби у повторному розгляді або аналізі зв'язків з попередніми кроками.

Документування характеристик стосовно динаміки задачі корисне з погляду проектування архітектури управління СППР, оскільки структура СППР повинна відображати основну динаміку задачі (наприклад, СППР, що проектується для ітеративних ситуацій із прийняття рішень, повинна мати замкнуту циклічну структуру). Крім того, деякі конкретні прийоми проектування СППР можуть бути застосовані тільки до ситуацій із прийняття рішень для конкретної динаміки задачі.

**3. Критерії вибору рішення.** Більшість рішень розглядаються по-своєму різними ОПР. Це пояснюється тим, що вони користуються різними критеріями для оцінювання конкретного вибору. Однією із складних задач, з якими змушені зустрічатися ОПР, є об'єднання різних критеріїв вибору на множині альтернатив (або навіть для однієї альтернативи). У багатьох випадках це проблема знань, у якій найкращі (або просто прийнятні) стратегії і правила вибору альтернативного критерію не зовсім чіткі або ж невідомі взагалі. Додаткові труднощі виникають у випадках, коли ці критерії носять числовий характер, у зв'язку з чим виникає необхідність виконання складних числових маніпуляцій у процесі їхнього об'єднання.

Багато систем підтримки прийняття рішень накладають кількісні критерії на самих ОПР, наприклад, у грошових, виробничих або часових одиницях. Визначення цих критеріїв вибору, що повинне об'єднати ОПР, а також визначення змісту такого об'єднання відіграють важливу роль з погляду забезпечення підтримки для цієї частини процедури прийняття рішень.

**4. Первинний (вихідний) процес.** Задачі прийняття рішень завжди мають відношення до реальних конкретних процесів, що мають детерміновані і стохастичні характеристики (або елементи). Оцінка потенційних дій у напрямку досягнення головної мети залежить, таким чином, від впливу обраної стратегії дій на конкретний процес. При цьому виникає два види проблем. Часто буває так, що наявної інформації про процес недостатньо для ефективного прогнозування його розвитку. Але навіть якщо обсяг інформації достатній, то завжди виникають труднощі з прогнозуванням розвитку процесу у часі і просторі. Опис цього аспекту процесу прийняття рішень повинен виходити з двох перспектив:

- зовнішньої або біхевіористичної перспективи, тобто, опису фізичних процесів, з якими буде взаємодіяти ОПР;
- внутрішньої або когнітивної перспективи, тобто, опису розуміння і сприйняття процесу і динаміки його розвитку з погляду ОПР.

**5. Інформаційна підтримка.** Сама істотна різниця між сучасною ОПР і ОПР попереднього покоління, полягає в об'ємі наявної інформації, що розглядається. Раніше багато рішень приймалися за наявності невеликих обсягів інформації. На сьогодні рішення приймаються за наявності таких обсягів інформації, що досить часто перевищують необхідний. Оскільки час, що надається для ухвалення рішення, завжди обмежений, то така ситуація може призводити до витоку важливої інформації. Можна виділити такі три види інформації, з яким має справу ОПР:

- вихідна інформація, що має безпосереднє відношення до ухвалення рішення і динамічно змінюється на кожній ітерації ухвалення рішення;
- інформація про параметри процесу, що має безпосереднє відношення до ухвалення рішення, але не змінюється динамічно на кожній ітерації

ухвалення рішення; значення параметрів можуть змінюватися в різних циклах процесу ухвалення рішення;

- вихідна інформація, яку одержує ОПР на основі перших двох видів інформації і яка спрямована на підтримку процесу ухвалення рішення в цілому, є безпосередньо частиною загальногорішення.

Розглянутий опис необхідний для можливої розробки і створення бази даних та/або знань для СППР.

**6. Проміжний аналіз.** Очевидно, що досвідчені ОПР (експерти) завжди привносять свої власні накопичені знання і досвід у процес прийняття рішень. Вони розкривають специфічні властивості ситуації з прийняття рішень, що можуть надати допомогу у виборі стратегії дій і вибрати, таким чином, оптимальне рішення. Досвідчений експерт знає, як виконати проміжний аналіз ситуації з ухвалення рішення і, завдяки цьому, використовує дуже специфічні проблемно-орієнтовані стратегії прийняття рішень. Такий проблемно-орієнтований проміжний аналіз може бути ефективно виконаний за допомогою допоміжних засобів, наприклад ЕОМ. Однак, це можливо тільки у тому випадку, коли проектувальник СППР знає, що представляє собою і з чого складається процес мислення експерта. Ідентифікація й опис цього процесу складають предмет проміжного аналізу.

**7. Представлення рішення.** Досвідчений експерт з прийняття рішень має свій «алгоритм» мислення і представлення задачі, а тому вирішує її швидко і ефективно. Фактично для всіх експертів характерна тенденція обмірковування тактичних ситуацій у досить абстрактній формі. Наприклад, досвідчений шахіст розмежовує шахівницю на «зони», що знаходяться під контролем чорних, білих або ж вони займають нейтральне положення. Таке представлення допомагає правильно планувати і робити ходи і зв'язувати їх з головною метою – поставити мат королю. Тому важливо спроектувати СППР (а особливо інтерфейс користувача) так, щоб вона відображала проблему у такому ж вигляді, як це робить експерт. Подібні абстрактні методи представлення проблеми виявляють своє втілення у візуальних образах і картинах. Визначення

цього розумового представлення проблеми експертом дає можливість знайти основу для створення комп'ютерної графіки в межах інтерфейсу користувача.

**8. Необхідність використання суджень.** Незалежно від того наскільки глибоко проаналізована проблема, може залишитися частина суджень (розумових висновків), які неможливо точно описати. Експерт виконує подібну розумову роботу, але не може дати точний опис того, як він це робить. Таку погано обумовлену і погано структуровану частину задачі прийняття рішень називають «судженням». Незважаючи на те, що більшість процесів, пов'язаних із формуванням суджень експерта, поки що перебувають за межами можливостей комп'ютера, існують методи, що допомагають реалізувати кількісні судження.

Розглянутий формат опису проблеми, що ґрунтується на протоколі декомпозиції задачі, корисний з погляду визначення вербальних характеристик ситуацій, що мають місце при прийнятті рішень.

**Другий етап.** Аналіз ситуації з прийняття рішення. Наведена вище методика опису процесу проектування СППР поєднує проблемно-орієнтований аналіз зовнішнього оточення, зв'язаного з прийняттям рішення, з когнітивним аналізом цілей і процесом мислення досвідченої ОПР. На наступному етапі проектування СППР аналізується опис ситуації з метою ідентифікації тих «вузьких (слабких) місць» процесу прийняття рішень, де необхідна допомога комп'ютера. Надалі необхідно визначити конкретні обчислювальні процедури для включення в комп'ютерну систему.

**Третій етап.** Визначення функцій системи підтримки прийняття рішень. Сучасні технології проектування СППР дають можливість використовувати відповідно шість функцій для підтримки прийняття рішень з боку ОПР:

1. **Моделювання процесу.** Використовуючи існуючі моделі реальних процесів (або створюючи нові), можна створювати підсистеми прогнозування їхнього подальшого протікання і підсистеми синтезу оптимальних рішень на основі поточних даних (спостережень).

2. **Моделювання критеріїв.** За допомогою математичних методів можна знайти математичний опис або правила для автоматичного об'єднання

атрибутів, що характеризують різні варіанти рішень, що знімає когнітивні обмеження ОПР.

3. **Інформаційний менеджмент.** Для збереження, читання та обробки інформації, даних, знань використовують сучасні комп'ютерні технології. Завдяки цьому значно розширюються можливості ОПР із прийняття рішень і обробки даних.

4. **Автоматизований і напівавтоматизований аналіз і логічний висновок.** Для часткової або повної автоматизації процесу формування логічного висновку необхідно використовувати методи штучного інтелекту і чисельні методи. Це дасть можливість підвищити якість результату і зменшити час на розв'язок подібної задачі.

5. **Способи підтримки представлення результатів.** Для того щоб реалізувати функції доступу до інших СППР, баз даних і знань, необхідно застосовувати засоби комп'ютерної графіки й інструментарій для обробки мов.

6. **Підвищення якості суджень.** З метою усунення систематичних помилок, що впливають з деяких кількісних евристичних суджень людини, необхідно впроваджувати статистичні та інші методи корегування результатів.

Наступним і останнім етапом функціонального проектування є вибір конкретних обчислювальних процедур для реалізації кожної функції СППР у рамках зовнішніх і когнітивних обмежень, що впливають на прийняття рішень, а також наявної групи експертів для проектування СППР.

**Четвертий етап.** Вибір технології (методів) для реалізації СППР у межах біхевіористичних і когнітивних обмежень. Функціональне проектування СППР представляє собою когнітивний інженерний процес, що містить у собі двоетапне узгодження даних, що надходять від конкретного опису ситуації з прийняття рішень, з технологічною базою системи підтримки прийняття рішень. Перший етап уже розглянуто вище.

Конкретні потреби у підтримці прийняття рішень вже виявлені на етапі визначення функціональних вимог. Потреби у підтримці прийняття рішень визначають функції, що повинна виконувати СППР. На наступному етапі конкретні потреби у підтримці рішень узгоджуються з індивідуальними

методами СППР на основі фізичних атрибутів конкретної ситуації і процесу прийняття рішень ОПР без допоміжних засобів. У такий спосіб відбувається перетворення потреб у підтримці прийняття рішень в функціональний проект.

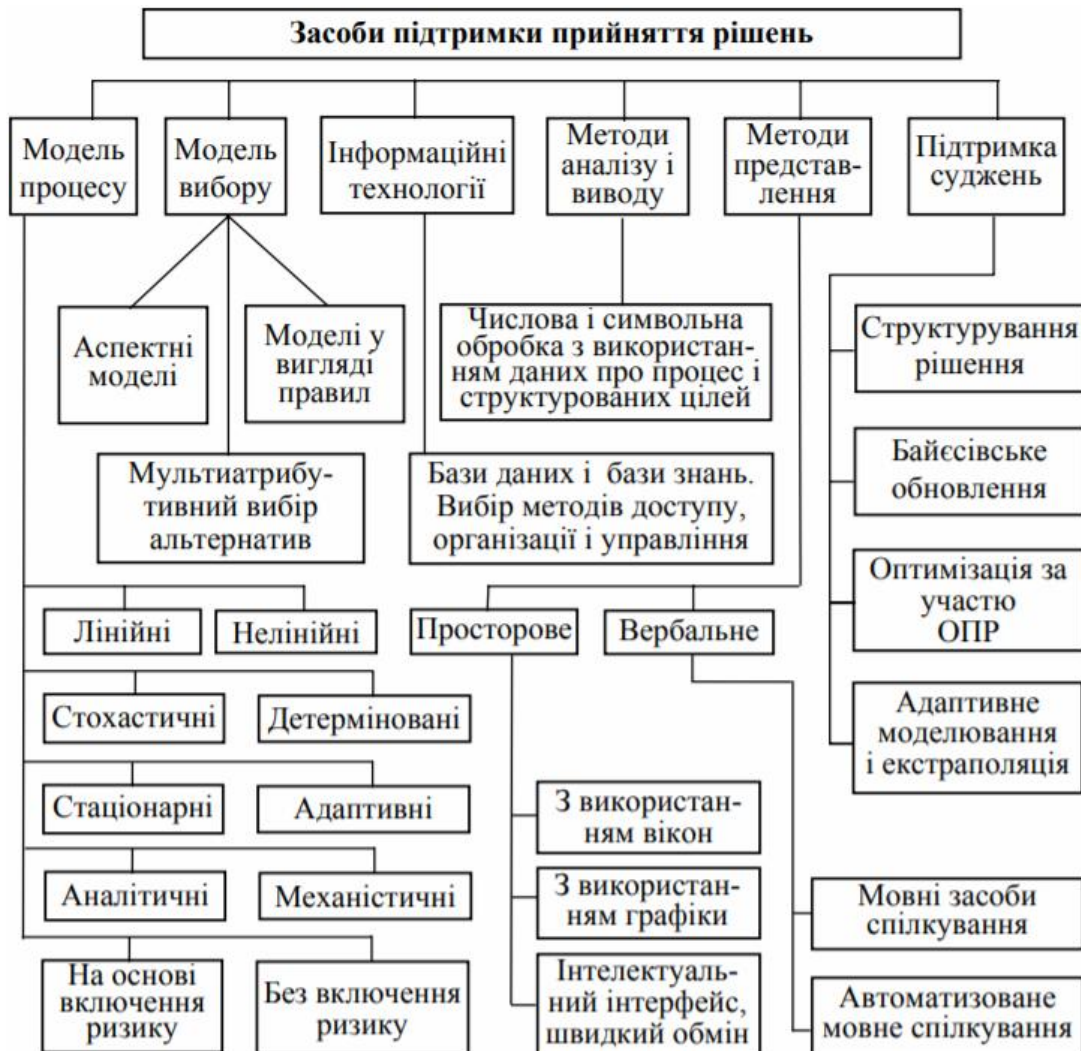


Рис. 2.5. Класифікація засобів підтримки прийняття рішень

Основним інструментом, який необхідно мати на цьому етапі, є чітке визначення і категоризація технологічної бази СППР, що, у свою чергу, ґрунтується на функціях і потребах, розглянутих вище. Шість видів математичного інструментарію, що використовується при проектуванні і реалізації СППР, наведено на рис. 2.5, тобто це такі види:

- математичні моделі реальних процесів, для управління або контролю яких створюється СППР;
- моделі вибору можливих альтернатив при пошуку рішення;

- інструментарій для інформаційного менеджменту;
- методи автоматизованого аналізу/логічного висновку;
- методи й інструментарій для представлення результатів;
- методи реалізації і підвищення якості суджень.

Кожна категорія складається з ряду конкретних методів, що можуть бути використані у конкретних випадках створення системи з метою реалізації функцій підтримки. Організація методів у межах кожної категорії ґрунтується на специфічних рисах або вимірах, що визначають ступінь їхньої продуктивності у конкретному випадку. Таким чином, класифікація, що представлена на рис. 2.5, не тільки дає перелік можливих методів, але й підкреслює ті конкретні риси проблеми з прийняття рішень, які необхідно розглядати при виборі конкретного методу для його реалізації в рамках СППР.

Очевидно, що сам по собі рис. 2.5 ще не дає достатньої підтримки для інформаційного проектування. Він визначає перелік наявних технологій для реалізації кожної бажаної функції СППР, але не визначає методики вибору конкретного методу або засобу порівняння з іншими. Для цього необхідно мати ще один інструмент – набір правил, що дозволяють реалізувати конкретні функції СППР. Ці правила повинні ґрунтуватися на даних, що характеризують ситуацію з ухвалення рішення і збираються шляхом декомпозиції/опису конкретної ситуації.

На етапі вибору методів кожна потреба у підтримці ухвалення рішення узгоджується з конкретним методом, наведеним на рис. 2.5. Іноді буває неможливо підібрати необхідний метод внаслідок високих вимог до результату або обмежень наявних технологій. Але, як свідчить практика, у більшості випадків можна підібрати щонайменше один метод.

Процес узгодження підбору методів необхідно виконувати для кожної конкретної задачі. Оскільки категорії, що використовуються для визначення конкретних потреб підтримки, пов'язані з характеристиками методів для підтримки рішень, то досить скористатися однією характеристикою для кожної з потреб підтримки. Хоча можливі такі випадки, коли необхідно розглядати також перехресні зв'язки. Деякі загальні правила узгодження конкретних

методів, представлених на рис. 2.5, з конкретними потребами підтримки прийняття рішень будуть розглянуті нижче.

### 2.3 Аналіз моделей і критеріїв для СППР

Існує декілька умов, що визначають необхідність використання моделі процесу. Очевидною умовою є те, що рішення повинне безпосередньо ґрунтуватися на первинному процесі, що аналізується. По-друге, поведження процесу необхідно прогнозувати в сильному або слабкому змісті. Процес розглядається як прогнозований у сильному змісті, якщо його внутрішня динаміка відома й описана математично. При цьому модель процесу може бути використана для його прогнозування поза межами встановлених «нормальних» умов функціонування. Поведження процесів, для яких неможливо створити точний математичний опис, також можна прогнозувати (екстраполювати) на основі спостережень, якщо вони є у достатньому обсязі. Такі процеси називають прогнозованими в «слабкому» змісті, тому що прогнозоване значення достовірне тільки для тих умов, що відповідають конкретному наборові даних. Як приклад «слабко прогнозованого» процесу можна навести процеси на товарному ринку. Економетричний аналіз свідчить, що агреговане поведження великого числа товарних транзакцій можна прогнозувати (екстраполювати) статистичними методами на основі минулих і поточних даних, не створюючи при цьому складних моделей динаміки.

Наступною передумовою використання моделі процесу є те, що необхідно мати дані, що характеризують його поведження в минулому. Відсутність цих даних можна замінити, в деякій мірі, експертними оцінками. Тобто при розробці моделі необхідно скористатися як знаннями експерта, так і числовими даними, якщо вони є в наявності. Правила вибору конкретної моделі процесу представлені на рис. 2.6.

Вибір конкретної моделі процесу ґрунтується на деяких додаткових характеристиках. Ці характеристики можна отримати за допомогою даних, зібраних на етапі декомпозиції задачі. Першою такою характеристикою є

присутність невизначеностей (невизначеностей у вихідних даних, в динаміці або в даних на виході). Якщо невизначеності мають місце, то необхідно вибрати стохастичну модель. Якщо ж невизначеності не грають істотної ролі, то процес можна розглядати як детермінований. Наступною характеристикою є бачення реального процесу з погляду ОПР [29].

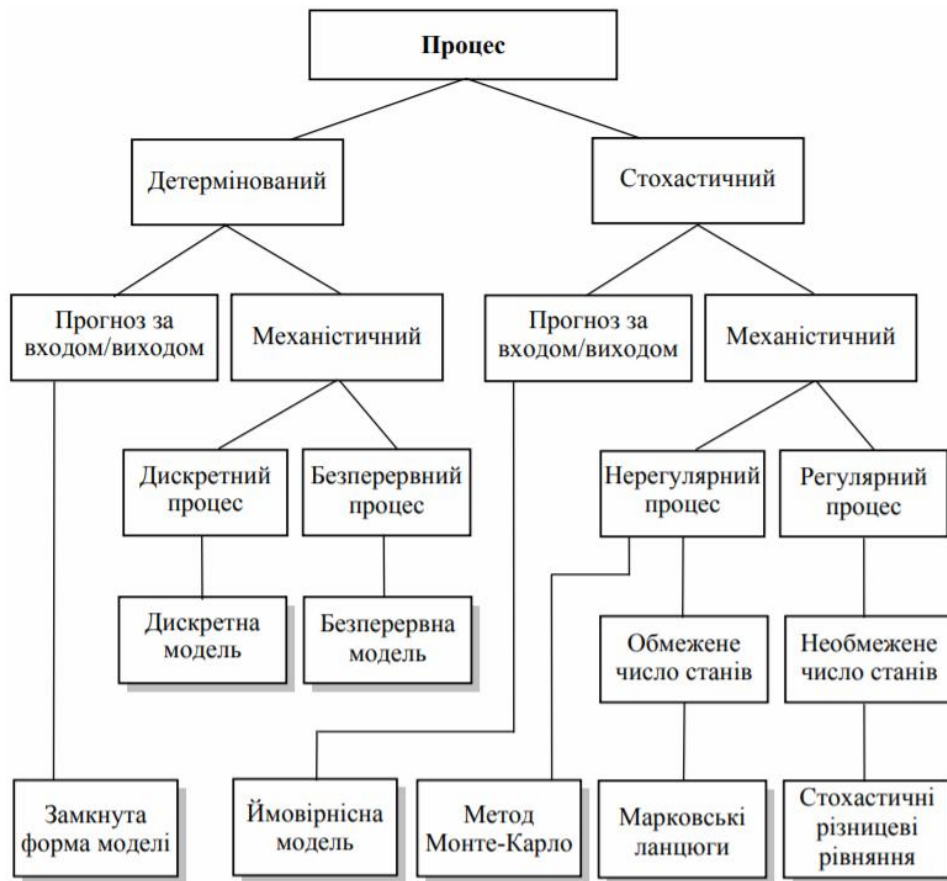


Рис. 2.6. Вибір моделі процесу при проектуванні СППР

ОПР має своє представлення про те, як необхідно прогнозувати процес. Якщо ОПР зацікавлена тільки в початковому положенні і кінцевому результаті, то таке прогнозування називають прогнозуванням на основі відношення вхід/вихід. Якщо ж ОПР бачить необхідність контролювати цілком протікання процесу (включаючи початкові умови і кінцеве положення), то таке прогнозування називають механістичним. Для детермінованого процесу і прогнозу на основі відношення вихід/вхід необхідно використовувати так звані замкнуті форми аналітичних моделей. Якщо ж процес стохастичний і

використовується прогноз на основі відношення вихід/вхід, то при проектуванні СППР необхідно скористатися ймовірнісною моделлю процесу.

Крім розглянутих характеристик необхідно визначити яку модель використовувати з погляду врахування протікання часу, тобто дискретну або неперервну. Очевидно, що більше труднощів пов'язано з аналізом стохастичних процесів. Якщо процес адекватно описується за допомогою 20-и або менше станів, то його вважають процесом з обмеженою кількістю станів. Ще однією характеристикою є регулярність процесу. Процес вважається регулярним, якщо він переходить з одного стану в інший через однакові інтервали часу. Інакше його класифікують як нерегулярний.

Якщо процес класифікується як стохастичний з обмеженим числом станів і регулярний, а форма прогнозування механістична, то при проектуванні необхідно вибрати модель на основі марковських ланцюгів і т. ін. Підхід до моделювання на основі методу Монте-Карло розглядають як «останню» можливість[30]. У даному випадку вважається, що зовсім немає можливості вивчити процес або зібрати обсяг даних, необхідний для побудови адекватної математичної моделі, але вважається, що є достатній обсяг обчислювальних ресурсів і часу для побудови подібної моделі.

## 2.4 Дослідження підходів до проектування СППР

За категорією класифікації – концептуальна модель (схема) – існують такі підходи до проектування: інформаційний, підхід, що ґрунтується на знаннях, та інструментальний підхід [31].

Основні компоненти СППР: інтерфейс “користувач – система”, база даних (БД) і база моделей. Інтерфейс включає засоби для управління базою даних, базою моделей генеруванням діалогу.

Структурну схему системи підтримки прийняття рішень зображено на рис. 2.7.

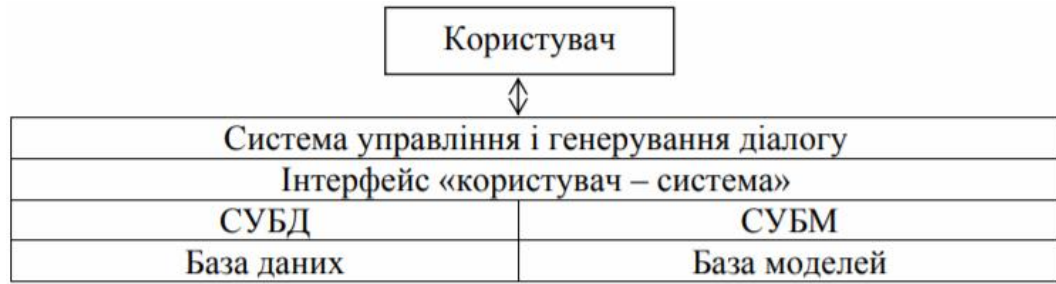


Рис. 2.7. Структурна схема традиційної СППР

Структурну схему СППР, яка ґрунтується на знаннях, зображено на рис. 2.8. Ця система складається з трьох взаємодіючих частин: мовна система (МС); система знань і система управління базою знань (СУБЗ), і підсистеми обробки (розв’язання) проблеми (проблемний процесор).

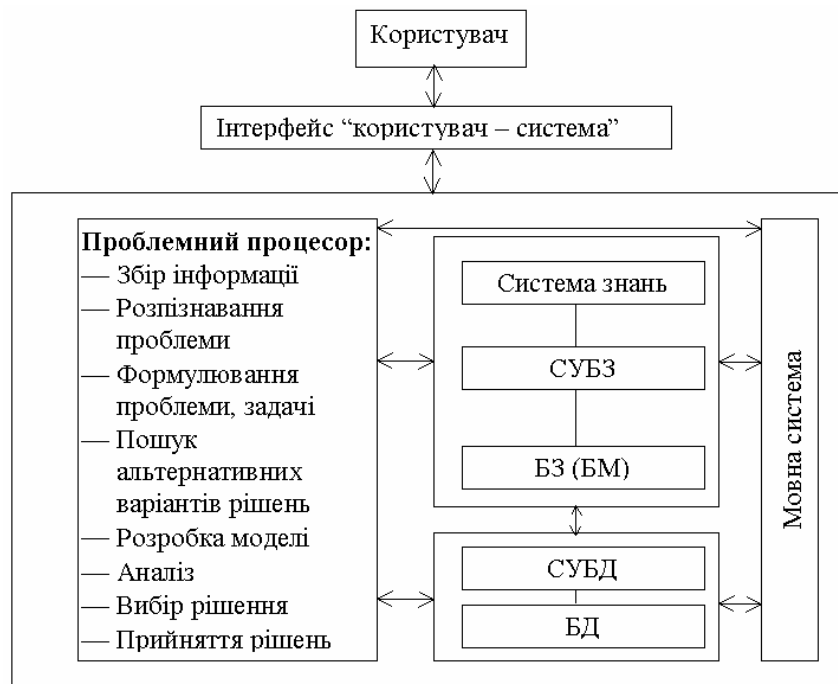


Рис. 2.8. Структурна схема СППР, яка ґрунтується на знаннях

Спеціалізовані СППР призначені для використання окремим користувачем або групами користувачів.

СППР-генератор – це пакет програмних засобів, який дає змогу створювати спеціалізовану СППР. Прикладом може бути інформаційна керуюча система, яка складається з різноманітних елементів: пошуку інформації, підготовки звітів, мови моделювання.

## 2.5 Структура та алгоритм роботи СППР

Ядром інтелектуальної системи є база знань, що містить в собі знання експертів щодо предметної області. Вона складається з двох компонентів:

- фактів – статичних даних, що введені екпертом;
- правил – набору інструкцій, що дозволяє отримати нові факти, на основі тих, що маютьяся.

База даних (робоча пам'ять) призначена для тимчасового зберігання початкових та проміжних даних, що є результатом спілкування інтелектуальної системи підтримки прийняття рішення з користувачем, що веде з нею діалог.

Підсистема спілкування забезпечує ведення діалогу з користувачем інтелектуальної системи, в ході якого система запитує в нього факти, необхідні для процесу міркування.

Машина логічного виводу являє собою механізм міркувань, що оперує з фактами та правилами з бази знань та даними, що отримані від користувача системи, та зберігаються в робочій пам'яті. Основною метою машини логічного виводу є реалізація стратегії вибору відповідного правила, факту.

Зокрема, вона керує чотирма процесами: вибором активних правил і фактів, їх зіставленням, вирішенням конфліктів, виконанням обраного зазначеного правила.

Підсистема пояснень необхідна для того, щоб дати можливість користувачу контролювати хід міркувань системи. У разі відсутності конструктивних пояснень щодо рекомендацій, користувач не матиме довіри до системи, і вона не матиме перспектив для використання.

Підсистема придбання знань автоматизує процеси наповнення системи знаннями, здійснювані екпертом. Лікар вносить свої знання в інтелектуальну систему, а саме в компонент придбання знань. Його основними функціями є: автоматизація процесу наповнення бази знань та актуалізація бази знань. Для функціонування підсистеми придбання знань розробляються алгоритми функціонування модуля накопичень знань та веб-інтерфейси користувачів, що дозволяють здійснювати операції маніпулювання зі знаннями.

При роботі з інтелектуальною системою підтримки прийняття рішення, користувач системи за допомогою мережі Інтернет заходить до веб-браузера, де вводить адресу системи підтримки прийняття рішення. При спрацюванні запиту, він бачить перед собою інтерфейс користувача, куди може ввести необхідні для аналізу дані.

Після введення, дані за допомогою веб-серверу передаються в робочу пам'ять інтелектуальної системи, звідки потрапляють до інших компонентів системи, в результаті чого формується результат роботи інтелектуальної системи, що потрапляє в робочу пам'ять та за допомогою веб-сервера відображається в інтерфейсі користувача.

Інтелектуальна система підтримки прийняття рішення працює в двох режимах: режимі придбання знань і в режимі вирішення задачі.

У режимі придбання знань спілкування з системою здійснює (через посередництво інженера по знаннях) експерт. В цьому режимі експерт, використовуючи компонент придбання знань, наповнює систему знаннями, які дозволяють інтелектуальній системі в режимі вирішення самостійно (без експерта) вирішувати завдання.

У режимі консультації спілкування з інтелектуальною системою здійснює кінцевий користувач, якого цікавить результат та (або) спосіб його отримання. В цьому режимі дані, що задані користувачем, після обробки їх діалоговим компонентом надходять в робочу пам'ять. Машина логічного виводу на основі вхідних даних з робочої пам'яті, загальних даних предметної області і правил з бази знань формує рішення задачі.

На рисунку 2.9 приведено структуру інтелектуальної системи підтримки прийняття рішення.

Узагальнений алгоритм функціонування інтелектуальної системи підтримки прийняття рішень наступний:

- 1) в інтелектуальну систему завантажуються список стандартних запитань, порядок і зміст яких є незмінним для кожного з користувачів;
- 2) користувач обирає одну з декількох відповідей на кожне запитання;

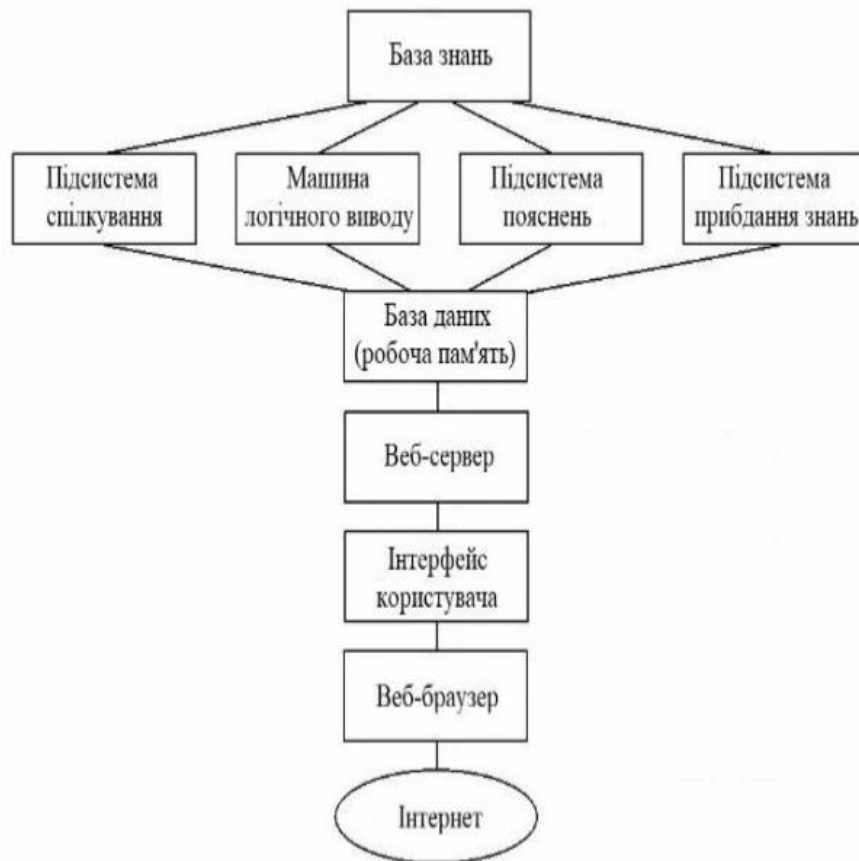


Рис. 2.9. Структура інтелектуальної системи підтримки прийняття рішення

3) на основі відповідей на стандартні запитання, визначається модуль бази знань, що містить групу наступних (уточнюючих) запитань, що належать обраній стандартній послідовності;

4) в систему завантажуються відповідна група уточнюючих запитань, визначається їх кількість та розраховується вага одного запитання;

5) користувачу відображуються запитання, він робить свій вибір:

- при відповіді «ні» вага запитання віднімається від загальної ваги, відбувається перехід до наступного запитання, обраного випадковим чином з усієї групи запитань;

- при відповіді «не знаю» або «так» до загальної ваги додається відповідно половина або повна вага запитання.

Відбувається перехід до наступного запитання, або п.б.;

б) вихідні дані, введені користувачем, зберігаються, коли загальна вага відповідей становить більше 80% (0.8);

7) проводиться зіставлення вихідних фактів з базою знань, визначається ідентифікатор проблеми, що отримала найбільшу вагу;

8) результат роботи системи у вигляді експертної поради відображається користувачеві.

Алгоритм роботи інтелектуальної системи підтримки прийняття рішення представлено на рисунку 2.10.



Рис. 2.10. Алгоритм роботи інтелектуальної системи підтримки прийняття рішення

## 2.6 Дослідження типів архітектури спеціалізованих СППР

Розглянемо текстово-орієнтовані системи підтримки прийняття рішень. Такі системи створювались і в базовій версії вони містили наступні компоненти:

- мовну систему (МС), що забезпечує функції взаємодії користувача з системою;
- базу знань та даних (БЗД);
- систему обробки даних та генерування результатів (СОДГР), яка виконує різноманітні маніпуляції над текстовою документацією і включає програмне забезпечення (ПЗ), що полегшує користувачеві складання запитів;
- систему представлення результатів (СПР).

Властивість текстової СППР – можливість гіпертекстової підтримки. Гіпертекст встановлює зв'язок між знаннями, які містяться в різних файлах тексту. При цьому кожний фрагмент тексту пов'язується з іншими фрагментами, які концептуально з ними пов'язані.

Наприклад, є фрагмент, де йде мова про потенційного конкурента. Цей фрагмент зв'язується з подібними фрагментами, де йде мова про інших конкурентів.

Дальше розглянемо СППР, які базуються на застосуванні бази даних (рис. 2.11).



Рис. 2.11. Структура СППР, що базується на основі БД

В такій системі виділяють три класи ПЗ:

- ПЗ для системи управління БД;
- спеціалізоване ПЗ, яке створюється для задоволення потреб користувача (включає, як правило, деяку логіку стосовно аналізу даних і формування відповідей на запити, а також необхідні обчислення: статистичні розрахунки, оцінювання параметрів моделей і прогнозів, порівняння отриманих результатів);
- інтерактивне ПЗ для обробки запитів.

Ще одним типом СППР, є системи які орієнтовані на використання електронних таблиць (ЕТ типу Excel).

Такі СППР мають такі характеристики:

- БЗД базується на основі файлів з таблицями, що заповнені описовими і процедурними знаннями.
- СОДГР може виконувати алгоритмічні процедури, тобто аналізу типу “що буде, якщо ...”.

Крім процедурних знань (формули у комірках) і описових знань (числа у комірках), таблиця може містити прості знання стосовно представлення результатів, а також лінгвістичні знання.

І на завершення розглянемо гібридні системи підтримки прийняття рішень. Якщо об'єднати декілька типів СППР, то отримаємо так звану гібридну систему (рис. 2.12).

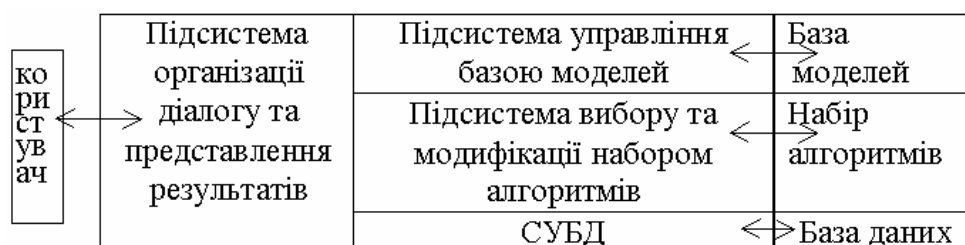


Рис. 2.12. Структура гібридної системи підтримки прийняття рішень

Очевидно, що такі системи є функціонально універсальнішими. Розглянемо базову архітектуру гібридної СППР (рис. 2.13).

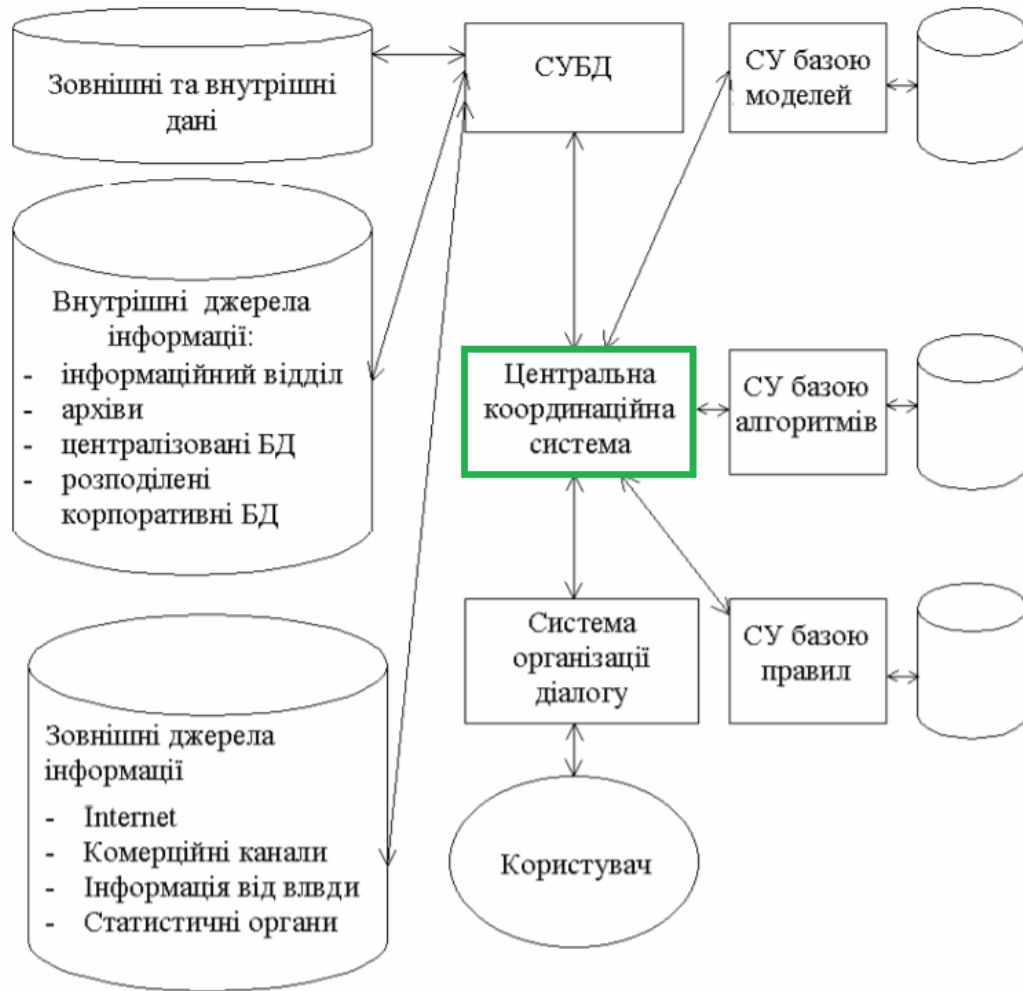


Рис. 2.13. Узагальнена архітектура гібридної СППР

За таким типом побудовано багато СППР:

- для управління повітряним рухом у аеропортах;
- СППР як радник в нештатних ситуаціях;
- довідки стосовно кадрів (персоналу);
- довідки щодо матеріально-енергетичних ресурсів;
- інформація про нештатні ситуації в минулому.

## 2.7 Аналіз моделей та технологій представлення даних у сховищах даних

Цінність і ефективність управлінських рішень залежить від якості отриманої інформації її точності і достовірності. Тому є дуже важливим

правильна підготовка вихідних даних. Одним із ефективних методів підготовки даних є консолідація інформації [35].

Консолідація це комплекс методів і процедур призначених для отримання даних з різних джерел, забезпечення необхідного рівня їх інформативності та якості, перетворення до єдиного формату, в якому вони можуть бути завантажені в сховище даних або аналітичну систему. В основі консолідації лежить процес збору і організації зберігання даних у вигляді найбільш оптимальному з точки зору їх обробки на конкретній аналітичній платформі. Основними критеріями оптимальності з погляду консолідації даних є: забезпечення високої швидкості доступу до даних, компактність зберігання, автоматична підтримка цілісності структури даних, контроль несуперечності даних. В процесі консолідації даних вирішуються наступні завдання: вибір джерел даних, оцінку якості даних, зберігання, очищення, перенесення в сховище даних.

Накопичення консолідованої інформації виконується у спеціальних базах даних або сховищах даних. Розглянемо організацію баз даних і сховищ даних .

В системах підтримки прийняття рішень широко застосовуються сховища даних які відрізняються від існуючих баз даних [36].

Сховища даних є предметно орієнтованими, інтегрованими, такими, що змінюються у часі і накопичують дані але їх не видаляють. На відміну від реляційних баз даних у сховищах даних, у яких інформація тільки накопичується, але не видаляється і не коригується, нормалізація таблиць не є необхідною. В організаційному відношенні, сховища даних є частиною процесу пошуку, зберігання і представлення даних для цілей підтримки прийняття рішень в управлінні. На рис. 2.14 показано місце сховища даних всередині ланцюгу обробки інформації системі прийняття рішень.

Для того, щоб сховище даних виконувало належні йому функцію - підтримки процесу аналізу даних та прийняття рішення, воно повинне відповідати вимогам, які сформульовані одним з авторів концепції сховищ даних - Ральфом Кімболлом:

- висока швидкість отримання даних зі сховища;

- автоматична підтримка внутрішньої несуперечності даних;
- можливість отримання і порівняння зрізів даних;
- наявність зручних засобів для перегляду даних в сховищі;
- забезпечення цілісності і достовірності даних.

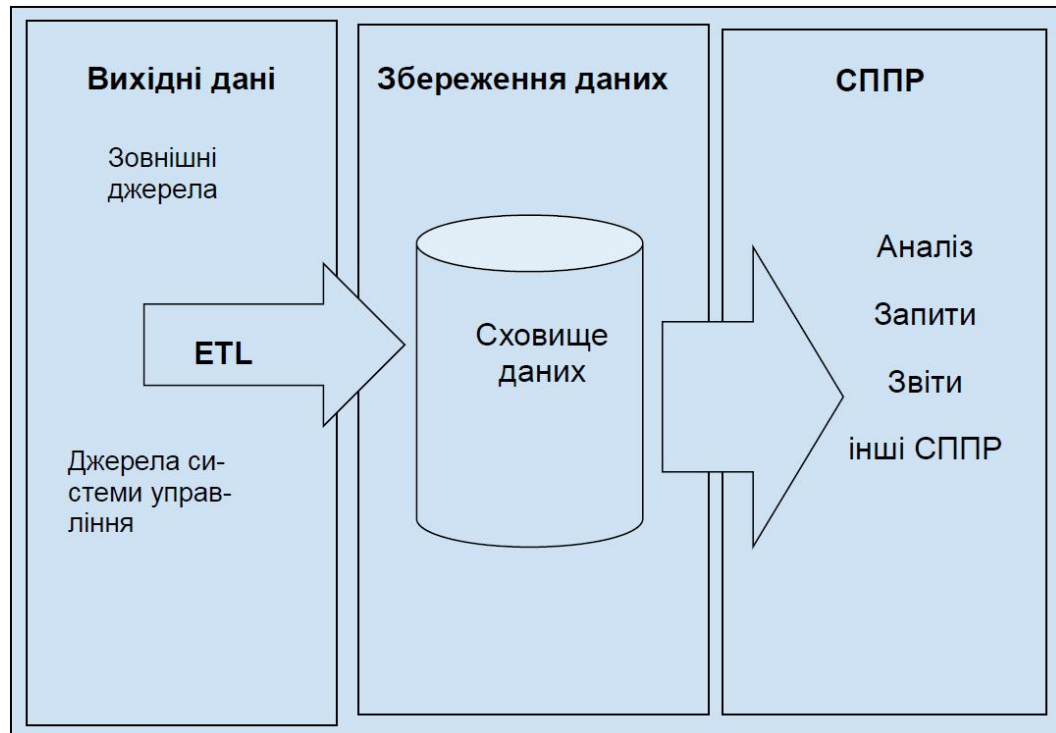


Рис. 2.14. Загальна модель зберігання даних

Для того, щоб задовольнити всі перелічені вимоги, для побудови і роботи сховища даних, як правило, використовується не одну програму, а систему, в яку входить декілька програмних продуктів. Одні з них є системою зберігання даних, інші - засоби їх перегляду, отримання, трансформації, завантаження, тощо.

Розробка і побудова сховищ даних - це досить недешеве і трудомістке завдання. Успішність впровадження сховища даних багато в чому залежить від рівня інформатизації бізнес-процесів, інформаційних потоків на підприємстві, об'єму і структури даних, вимог до швидкості виконання запитів і періодичності оновлення сховища, характеру аналітичних завдань тощо. З метою наблизити сховище даних до умов і специфіки конкретної організації в

даний час розроблено декілька архітектур сховищ даних: багатовимірні, реляційні, гібридні і віртуальні.

Багатовимірна модель даних, яка лежить в основі побудови багатовимірних сховищ даних, оснований на концепції багатовимірних кубів або гіперкубів. Вони являють собою впорядковані багатовимірні масиви, які також часто називають OLAP-кубами (OLAP означає On-line Analytical Processing - оперативна аналітична обробка) [37]. Сховища даних побудованих на основі багатовимірності називають MOLAP (Multidimensional OLAP). Технологія OLAP - це методика оперативного отримання потрібної інформації з великих масивів даних і формування відповідних звітів.

Суть багатовимірного представлення даних полягає в наступному. Більшість реальних бізнес-процесів описуються великим набором показників, властивостей, атрибутів. Наприклад, для опису процесу продажу можуть знадобитися такі відомості, як найменування товарів або їх груп, інформація про постачальника і покупця, місто, де проводилися продажі, про ціни, кількості і суми проданих товарів. Окрім цього, для відстеження процесу в часі потрібно ввести такий атрибут, як дата та час. Якщо зібрати всю цю інформацію в таблицю, то вона виявиться складною для візуалізації та аналізу. Тому зберігання і обробка такої багатовимірної інформації в одній двовимірній таблиці є не ефективною.

В основі багатовимірного представлення даних лежить їх розділення на дві групи - виміри і факти.

Багатовимірний куб можна розглядати як систему координат, осями якої є виміри - дата, товар, покупець. По осях відкрадватимуться значення вимірів - дати, найменування товарів, назви фірм-покупців, ПІБ фізичних осіб. В такій системі кожному набору значень вимірів (наприклад, дата-товар-покупець) відповідатиме комірка, в якій можна розмістити числові показники (тобто факти), пов'язані з даним набором. Таким чином, між об'єктами бізнес-процесу та їх числовими характеристиками буде встановлений однозначний зв'язок.

Таким чином, інформація в багатовимірному сховищі даних є логічно цілісною. Це вже не просто набори строкових і числових значень, які у випадку

з реляційною моделлю потрібно отримувати з різних таблиць, а цілісні структури типу «кому, що і скільки було продано на певний момент часу. Переваги багатовимірного підходу є наступними [38]:

- представлення даних у вигляді багатовимірних кубів є більш наочним, ніж сукупність нормалізованих таблиць реляційної моделі, структуру якої представляє тільки адміністратор БД;
- ширші можливості побудови аналітичних запитів до системи, що використовує БСД;
- використання багатовимірної моделі в деяких випадках дозволяє різко зменшити час пошуку в БСД, забезпечуючи виконання аналітичних запитів практично в режимі реального часу. Це пов'язано з тим, що агреговані дані обчислюються заздалегідь і зберігаються в багатовимірних кубах разом з деталізованими. Тому витратити час на обчислення агрегатів при виконанні запиту вже не потрібно;
- формування навіть найскладніших аналітичних запитів до БСД здійснюється без використання SQL з боку користувача.

На підставі цього можна зробити висновок, що використання систем збереження, в основі яких лежить багатовимірне представлення даних, доцільно в тих випадках, коли об'єм даних порівняно невеликий, а сама багатовимірна модель має стабільний набір вимірів.

В принципі, багатовимірний куб може бути реалізований і за допомогою звичайної реляційної моделі. В цьому випадку має місце емуляція багатовимірного уявлення сукупністю плоских таблиць. Такі системи отримали назву ROLAP - Relational OLAP.

Реляційна база даних (relational data base) - це сукупність відносин, що містять всю інформацію, яка зберігається в базі. Фізично це виражається в тому, що інформація зберігається у вигляді двовимірних таблиць, зв'язаних за допомогою ключових полів.

Застосування реляційної моделі при створенні сховище даних часто дозволяє отримати ряд переваг, особливо щодо ефективності роботи з великими

масивами даних. На основі РСД будуються ROLAP-системи (Relational OLAP). В основі технології РСД лежить принцип, відповідно до якого виміри зберігаються в плоских таблицях так само, як і в звичайних реляційних СУБД, а факти (дані, що агрегуються) в окремих спеціальних таблицях цієї ж бази даних. При цьому таблиця фактів є «основою» для пов'язаних з нею таблиць вимірів. Вона містить кількісні характеристики об'єктів і подій, сукупність яких будуть аналізуватися в майбутньому [40].

Як MOLAP, так і ROLAP моделі сховищ даних мають свої переваги та недоліки. Наприклад, багатовимірна модель забезпечує високу швидкість реакції на запит, але не дозволяє ефективно управляти такими великими об'ємами даних як реляційна модель. Тоді логічно було б використовувати модель даних, яка була б комбінацією реляційної і багатовимірної моделі і дозволяла б поєднувати високу продуктивність, характерну для багатовимірної моделі, і можливість зберігати великі масиви даних, властиву реляційній моделі. Така модель, що поєднує в собі принципи реляційної і багатовимірної моделей, отримала назву гібридною або HOLAP - Hybrid OLAP. Сховища даних, побудовані на основі HOLAP, називаються гібридними сховищами даних.

Головним принципом побудови гібридного сховища даних є те, що деталізовані дані зберігаються в реляційній структурі (ROLAP), що дозволяє зберігати великі об'єми даних, а агреговані - в багатовимірній (MOLAP), що дозволяє збільшити швидкість виконання запитів (оскільки при виконанні аналітичних запитів вже не потрібно обчислювати агрегати).

Для кращого розуміння сказаного, наведемо наступний приклад. Нехай облікова система, встановлена в супермаркеті, обслуговуючому щодня тисячі покупців. При цьому максимальному рівню деталізації даних відповідає покупка по одному чеку, в якому вказується загальна сума покупки, найменування або коди придбаних товарів і сума по кожному товару. Оперативна інформація, що складається з деталізованих даних, консолідується в реляційній структурі сховища даних. Інтерес з погляду аналізу представляють узагальнені (агреговані) дані, наприклад, по групах товарів, відділах або деяких

періодах. Тому початкові деталізовані дані агрегуються і обчислені агрегати зберігаються в багатовимірній структурі гібридного сховища даних [42].

Якщо дані облікової системи мають великий об'єм (декілька тисяч транзакцій в день і більше) і високий ступінь деталізації, а використовуються для аналізу в основному узагальнені дані, то гібридна архітектура сховища виявляється найбільш підходящою. Однак зрозуміло, що крім позитивних рис, гібридна модель має і недоліки (табл. 2.1.)

Таблиця 2.1

#### Переваги та недоліки гібридної моделі сховища даних

Переваги	Недоліки
<ul style="list-style-type: none"> <li>- Реляційна структура формус стійкі і несуперечливі таблиці і зв'язки між ними;</li> <li>- Оскільки реляційне сховище підтримує актуальність і коректність даних, воно забезпечує надійний транспортний рівень для передачі інформації в багатовимірне сховище.</li> </ul>	<ul style="list-style-type: none"> <li>- Ускладнене адміністрування, що зумовлено більш складною процедурою поповнення даними такого сховища. Потрібно узгоджувати зміни в реляційній та багатовимірній частині сховища.</li> </ul>

Більшість програмних рішень для створення сховищ даних використовують саме гібридну модель. Так, наприклад, корпорація Oracle в своїх розробках використовує ROLAP для збереження даних в сховищі, вітрини даних побудовані на базі MOLAP.

#### 2.7.1 Аналіз даних та формування OLAP-кубів

Реалізації системи підтримки прийняття рішень для вирішення задач планування багатомножинного харчового підприємства передбачає наступні етапи:

1. Проектування схеми сховища даних з урахуванням всіх параметрів даних для аналізу собівартості продукції.

2. Проектування та побудова OLAP-кубів, наповнених інформацією, необхідною для аналізу факторів, які впливають на собівартість продукції.

3. Підготовка даних для формування прогнозних рішень на основі алгоритмів інтелектуального аналізу даних Data Mining.

4. Розробка рішень щодо візуалізації даних користувачеві в OLAP-клієнті для інформаційної підтримки прийняття управлінських рішень планування.

Реалізацію системи інформаційної підтримки для аналізу та прийняття рішень планування собівартості продукції харчового підприємства.

Проаналізувавши переваги та недоліки основних СУБД, які присутні на українському ринку, в якості середовища для збереження даних було обрано клієнт-серверну СУБД MS SQL Server.

Проаналізувавши роботу СППР, можна виділити її наступні основні функції:

- формування OLAP-кубів, наповнених інформацією, необхідною для аналізу факторів, які впливають на собівартість продукції;
- моніторинг собівартості продукції та інших показників для виявлення їх відхилень від планового рівня;
- пошук прихованих закономірностей на основі алгоритмів інтелектуального аналізу даних Data Mining для формування рекомендацій щодо коригування структури асортименту продукції;
- прогнозування та оцінку ризику прийняття рівня собівартості продукції;
- прогнозування попиту на продукцію;
- побудову графіку беззбитковості;

Джерелом даних для наповнення OLAP-кубів інформацією є розроблене в Microsoft SQL Server Management Studio сховище даних.

Формування вихідних результатів у формі звітів, зведених таблиць, графіків та діаграм здійснюється на клієнтському комп'ютері.

## 2.8 Висновки до розділу

На етапі вибору методів кожна потреба у підтримці ухвалення рішення узгоджується з конкретним методом. Іноді буває неможливо підібрати

необхідний метод внаслідок високих вимог до результату або обмежень наявних технологій. Але, як свідчить практика, у більшості випадків можна підібрати щонайменше один метод.

Процес узгодження підбору методів необхідно виконувати для кожної конкретної задачі. Оскільки категорії, що використовуються для визначення конкретних потреб підтримки, пов'язані з характеристиками методів для підтримки рішень, то досить скористатися однією характеристикою для кожної з потреб підтримки. Хоча можливі такі випадки, коли необхідно розглядати також перехресні зв'язки

## РОЗДІЛ 3

# МОДЕЛІ ТА МЕТОДИ ОБРОБКИ ДАНИХ ТА РЕАЛІЗАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЗАСОБАМИ OLAP І DATA MINING

### 3.1 Дослідження та описання алгоритмів, на яких базується система обробки даних та генерування результатів

Розробка системи обробки даних та генерування результатів (СОДГР) є головною системою (ядром) СППР з точки зору її основного призначення – підтримки прийняття рішень при розв’язанні задач в обраній прикладній області. Розглянемо функції СОДГР на прикладі СППР при прогнозуванні динаміки часових рядів [43].

СОДГР приймає коректні запити від мовної системи (МС) і виконує такі дії (які задаються запитом):

- поповнює (при необхідності) базу даних і знань;
- вибирає алгоритм (алгоритми) обробки даних з бази даних і знань;
- застосовує вибраний алгоритм до даних, що містяться в базі даних, з метою побудови математичної моделі, обчислення прогнозу чи керуючих дій або виконання інших функцій СППР;
- використовує критерії адекватності моделі з метою визначення ступеня адекватності побудованої моделі;
- застосовує критерії якості прогнозу (або критерії якості керування) для визначення кращого прогнозованого значення або кращої траєкторії оптимального керування вибраним процесом;
- при необхідності реалізує функцію ретроспективного аналізу результатів з метою порівняння останнього отриманого результату (прогнозування, керування, планування і т.ін.) з результатами, які були отримані раніше; таке порівняння дає змогу визначити існування подібних ситуацій в минулому і теперішньому часі;

- передає отримані результати обчислень в систему представлення результатів, яка представляє отриманий результат у формі, заданій користувачем;
- зберігає, при необхідності, частину отриманих результатів в короткостроковій пам'яті для подальшого використання в поточній сесії використання СППР.

При проектуванні СОДГР необхідно вибрати та описати всі обчислювальні алгоритми та алгоритми прийому/передачі даних, які будуть використовуватись в процесі підтримки прийняття рішень.

Якщо СППР призначена для прогнозування, то це будуть такі алгоритми:

- алгоритми попередньої обробки даних (логарифмування, нормування, фільтрація, заповнення пропусків, обробка великих імпульсних значень);
- оцінювання параметрів (коефіцієнтів) математичних та/або статистичних моделей;
- алгоритми обчислення статистичних параметрів, які характеризують якість (адекватність) побудованої моделі;
- алгоритми обчислення оцінок прогнозованих значень (на основі рівнянь АР, АРКС, АРІКС, АРУГ, УАРУГ множинної регресії; за допомогою МГВА; за допомогою фільтра Калмана; за методом подібних траєкторій; методами експоненціального згладжування; нечітких множин; нейронних мереж і т.ін.);
- алгоритми обчислення статистичних показників якості прогнозів;
- алгоритм(и) (або правила) вибору кращої моделі та кращого прогнозу на основі розрахованих показників адекватності моделі та якості прогнозу;
- алгоритми формування бази результатів для проведення ретроспективного аналізу;
- алгоритми поповнення бази даних і знань.

Очевидно, що проектування СППР іншого типу потребує використання алгоритмів іншого функціонального призначення. Наприклад, якщо проектується СППР текстового типу, то СОДГР повинна містити алгоритми пошуку текстової інформації в базі знань, сортування у відповідності до запиту, забезпечення гіпертекстової підтримки, формування звітів і та інше.

### **3.2 Дослідження аналізу даних технологією Data Mining**

Для того щоб провести автоматичний аналіз даних, використовується Data Mining (здобич (витягання) знань). Це технологія інтелектуального аналізу даних з метою виявлення прихованих закономірностей у вигляді значущих особливостей, кореляцій, тенденцій і шаблонів. Сучасні системи «здобичі» даних використовують засновані на методах штучного інтелекту засоби уявлення і інтерпретації, що і дозволяє знаходити розчинену в терабайтних сховищах не очевидну, але вельми цінну інформацію. Фактично, ми говоримо про те, що в процесі Data mining система не відштовхується від наперед висунутих гіпотез, а пропонує їх сама на основі аналізу.

Існує безліч визначень Data Mining, але в цілому вони співпадають у виділенні чотирьох основних ознак:

- раніше не були визначені;
- нетривіальні;
- корисні;
- доступні для інтерпретації і необхідні для ППР в різних сферах діяльності.

Специфіка сучасних вимог до продуктивної переробки інформації:

- дані мають необмежений обсяг;
- дані є різномірними (кількісними, якісними, текстовими);
- результати — конкретні та зрозумілі;
- інструменти для обробки «сирих даних» — прості у використанні.

Традиційна математична статистика, яка довгий час претендувала на роль основного інструменту аналізу даних, не відповідала новим проблемам. Головна причина — концепція усереднювання по вибірці, що тягне за собою операції над фіктивними величинами. Методи математичної статистики виявилися корисними, головним чином, для перевірки наперед сформульованих гіпотез і для «грубого розвідувального аналізу», який є основою оперативної аналітичної обробки даних OLAP.

Основа сучасної технології Data Mining — концепція шаблонів (pattern), що відображають фрагменти багатоаспектних взаємостосунків даних. Цими шаблонами є закономірності, властиві підвибіркам даних, які можуть бути компактно виражені у формі, зрозумілій людині. Пошук шаблонів проводиться методами, не обмеженими рамками апріорних припущень про структуру вибірки і видом розподілів значень аналізованих показників. Причини популярності Data Mining:

- стрімке накопичення даних (рахунок йде вже на екзабайти);
- загальна комп'ютеризація бізнес-процесів;
- проникнення Інтернет у всі сфери діяльності;
- прогрес в області інформаційних технологій: вдосконалення СУБД і сховищ даних;
- прогрес в області виробничих технологій: стрімке зростання продуктивності комп'ютерів, об'ємів накопичувачів, впровадження Grid-систем.

Алгоритми, які використовуються в Data Mining, вимагають великої кількості обчислень. Раніше це було стримуючим чинником широкого практичного застосування Data Mining, проте сьогоднішнє зростання продуктивності сучасних процесорів зняло гостроту цієї проблеми. Data Mining — міждисциплінарна галузь, що виникла і розвивалася на базі таких наук, як прикладна статистика, штучний інтелект, розпізнавання, теорія БД (рис. 3.1).

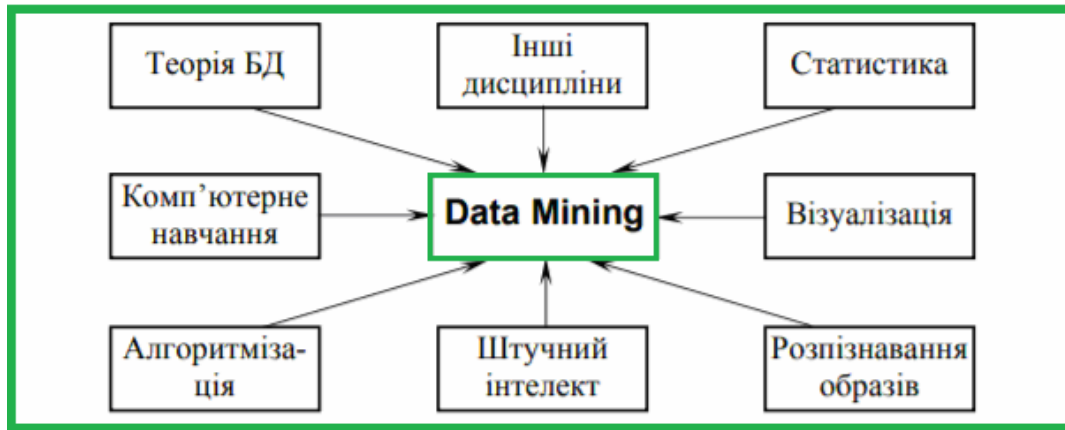


Рис. 3.1. Data Mining як міждисциплінарна галузь

До методів і алгоритмів Data Mining належать: штучні нейронні мережі, дерева рішень, метод опорних векторів, методи кластерного аналізу, байесові мережі, лінійна регресія, кореляційно-регресійний аналіз; пошук асоціативних правил, тощо.

Основні задачі Data Mining:

- Класифікація (Classification). Найпростіша і поширеніша задача Data Mining;
- Кластеризація (Clustering). Логічне продовження ідеї класифікації;
- Асоціація (Associations);
- Послідовність (Sequence);
- Прогнозування (Forecasting);
- Візуалізація (Visualization, Graph Mining). Створюється графічний образ аналізованих даних. Для вирішення цієї задачі використовуються графічні методи, які показують наявність закономірностей у даних.
- Підведення підсумків (Summarization). Опис конкретних груп об'єктів з аналізованого набору даних та ін.

### 3.3 Реалізація архітектури СППР з використанням OLAP-технологій і Data Mining

Сучасні СППР виникли у результаті злиття управлінських інформаційних систем і систем управління базами даних, як системи, що максимально

пристосовані до розв'язування задач щоденної управлінської діяльності, і є інструментом, щоб надати допомогу тим, хто вирішує (робить вибір). За допомогою СППР може проводитись вибір рішень у певних неструктурованих і слабо структурованих задачах, у тому числі й тих, що мають багато критеріїв.

На рис. 3.2 представлена загальна архітектура СППР. З організаційної точки зору така система включає такі компоненти: підсистему вводу, підсистему зберігання інформації та підсистему аналізу.

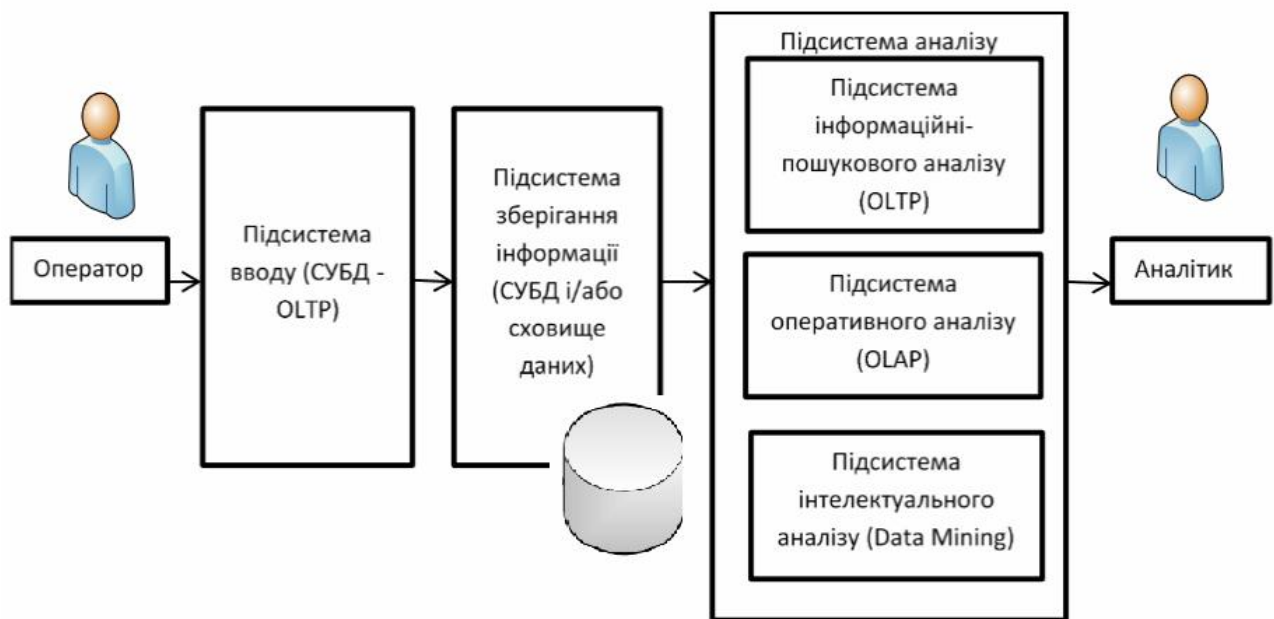


Рис. 3.2. Загальна архітектура системи підтримки прийняття рішень

У підсистемі зберігання інформації використовують сучасні СУБД і концепцію сховищ даних.

Підсистема аналізу може включати:

1) підсистему інформаційно-пошукового аналізу на базі реляційних СУБД і статичних запитів з використанням мови SQL;

2) підсистему оперативного аналізу, для її реалізації застосовується технологія оперативної аналітичної обробки даних OLAP (Online Analytical Processing, аналітична обробка в реальному часі), тобто технологія обробки інформації, що дозволяє швидко отримувати відповіді на багатовимірні аналітичні запити;

3) підсистему інтелектуального аналізу, яка реалізує методи і алгоритми Data Mining для видобування знань.

Компоненти СППР виконують такі основні питання: накопичення даних, ефективного завантаження даних із кількох незалежних джерел та аналізу даних.

OLAP – технологія для оперативної аналітичної обробки даних, яка використовує методи і засоби для збору, зберігання і аналізу багатовимірних даних з метою підтримки процесів прийняття рішень.

Основне призначення OLAP-систем – підтримка аналітичної діяльності, довільних запитів користувачів-аналітиків. Мета OLAP-аналізу – перевірка гіпотез.

OLAP-системи надають аналітику засоби перевірки гіпотез при аналізі даних. При цьому основним завданням аналітика є генерація гіпотез. Він вирішує її, ґрунтуючись на своїх знаннях і досвіді. Однак знання є не тільки у людини, але і в накопичених даних, які піддаються аналізу. Такі знання часто називають "прихованими", так як вони містяться в гігабайтах і терабайт інформації, які людина не в змозі дослідити самотійно. У зв'язку з цим існує висока ймовірність пропустити гіпотези, які можуть принести значну вигоду.

Для виявлення прихованих знань необхідно застосовувати спеціальні методи автоматичного аналізу, за допомогою яких доводиться практично здобувати знання з "завалів" інформації. За цим направлення міцно закріпився термін видобуток даних або Data Mining.

Data Mining – дослідження і виявлення "машиною" (алгоритмами, засобами штучного інтелекту) в сирих даних прихованих знань, які раніше не були відомі, нетривіальні, практично корисні, доступні для інтерпретації людиною.

Властивості для визначення знань:

- Знання повинні бути нові, раніше невідомі. Витрачені зусилля на відкриття знань, які вже відомі користувачеві, не окупаються. Тому цінність представляють саме нові, раніше невідомі знання.

- Знання повинні бути нетривіальні. Результати аналізу повинні відображати неочевидні, несподівані закономірності в даних, які складають так звані приховані знання. Результати, які могли б бути отримані більш простими способами (наприклад, візуальним переглядом), не виправдовують залучення потужних методів Data Mining.

- Знання повинні бути практично корисні. Знайдені знання повинні бути застосовні, в тому числі і на нових даних, з досить високим ступенем достовірності. Корисність полягає в тому, щоб ці знання могли принести певну вигоду при їх застосуванні.

- Знання повинні бути доступні для розуміння людині. Знайдені закономірності повинні бути логічно пояснюючими, в іншому випадку існує ймовірність, що вони є випадковими. Крім того, визначені знання повинні бути представлені в зрозумілому для людини вигляді.

Методи Data Mining допомагають вирішити багато завдань, з якими стикається аналітик. З них основними є: класифікація, регресія, пошук асоціативних правил і кластеризація. Опис основних задач аналізу даних.

1. Завдання **класифікації** зводиться до визначення класу об'єкта по його характеристиках. Необхідно зауважити, що в цьому завданні безліч класів, до яких може бути віднесений об'єкт, відомо заздалегідь.

2. Завдання **регресії** подібно задачі класифікації дозволяє визначити за відомими характеристиками об'єкта значення деякого його параметра. На відміну від завдання класифікації значенням параметра є не кінцевою множиною класів, а множиною дійсних чисел.

3. При **пошуку асоціативних правил** метою є знаходження частих залежностей (або об'єднань) між об'єктами або подіями. Знайдені залежності представляються у вигляді правил і можуть бути використані як для кращого розуміння природи аналізованих даних, так і для передбачення появи подій.

4. Завдання **кластеризації** полягає в пошуку незалежних груп (кластерів) і їх характеристик у всьому безлічі аналізованих даних. Вирішення цього завдання допомагає краще зрозуміти дані. Крім того, угруповання однорідних об'єктів дозволяє скоротити їх число, а отже, і полегшити аналіз.

Для виявлення знань в даних недостатньо просто застосувати методи Data Mining, хоча, безумовно, цей етап є основним в процесі інтелектуального аналізу. Весь процес складається з кількох етапів.

Розглянемо основні з них, щоб продемонструвати, що без спеціальної підготовки аналітика методи Data Mining самі по собі не вирішують існуючих проблем.

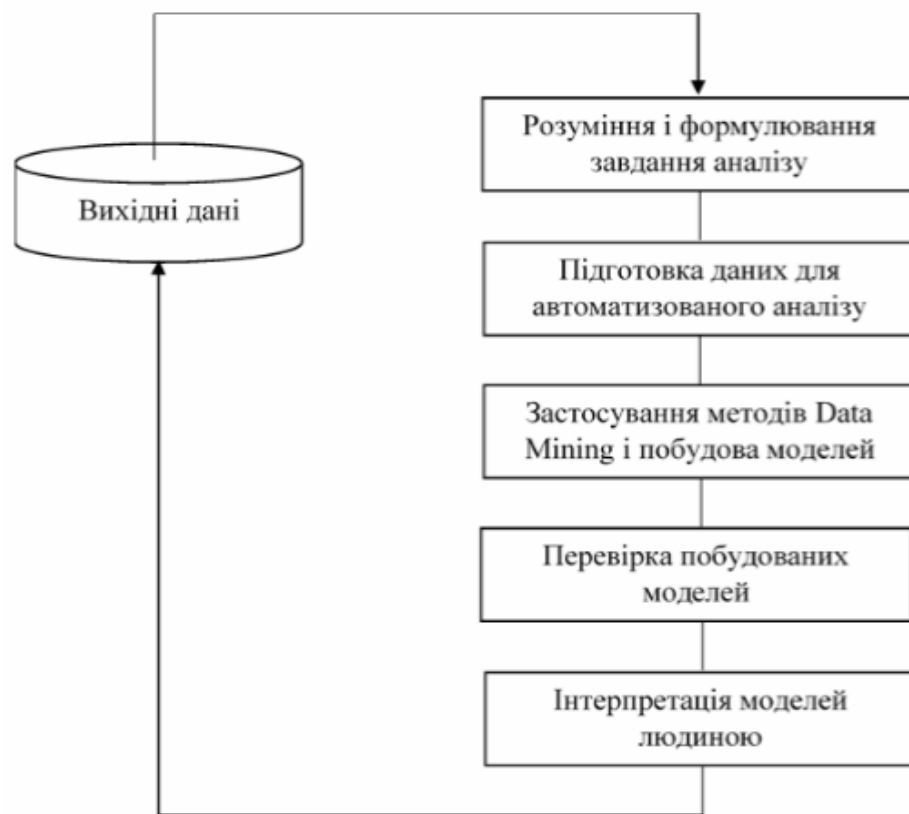


Рис. 3.3. Етапи інтелектуального аналізу даних

Отже, весь процес можна розбити на наступні етапи (рис. 3.3):

- 1) Розуміння і формулювання завдання аналізу.
- 2) Підготовка даних для автоматизованого аналізу (препроцесінг).
- 3) Застосування методів Data Mining і побудова моделей.
- 4) Перевірка побудованих моделей.
- 5) Інтерпретація моделей людиною.

На першому етапі виконується осмислення поставленої задачі і уточнення цілей, які повинні бути досягнуті методами Data Mining.

Другий етап полягає у приведенні даних до форми, придатної для застосування конкретних методів Data Mining.

Третій етап – це власне застосування методів Data Mining. Сценарії цього застосування можуть бути самими різними і можуть включати складну комбінацію різних методів, особливо якщо використовуються методи дозволяють проаналізувати дані з різних точок зору.

Наступний етап – перевірка побудованих моделей. Дуже простий і часто використовуваний спосіб полягає в тому, що всі наявні дані, які необхідно аналізувати, розбиваються на дві групи. Як правило, одна з них більшого розміру, інша – меншого. На більшій групі, застосовуючи ті чи інші методи Data Mining, отримують моделі, а на меншій – перевіряють їх. За різницею в точності між тестової і навчальної групами можна судити про адекватність побудованої моделі.

Останній етап – інтерпретація отриманих моделей людиною з метою їх використання для прийняття рішень, додавання одержаних правил і залежностей в бази знань і т. д. Цей етап часто має на увазі використання методів, які перебувають на стику технології Data Mining і технології експертних систем. Від того, наскільки ефективним він буде, в значній мірі залежить успіх вирішення поставленого завдання. Цим етапом завершується цикл Data Mining. Остаточна оцінка цінності видобутого нового знання виходить за рамки аналізу, автоматизованого або традиційного, і може бути проведена тільки після втілення в життя рішення, прийнятого на основі здобутого знання, після перевірки нового знання практикою. Дослідження досягнутих практичних результатів завершує оцінку цінності видобутого засобами Data Mining нового знання.

Інструменти Data Mining можуть самостійно знаходити закономірності і будувати гіпотези про взаємозв'язки. Інтеграція технологій OLAP і Data Mining розширює функціональність СППР. Ці два види аналізу мають бути тісно поєднані, щоб інтегрована технологія могла забезпечувати одночасно багатовимірний доступ і пошук закономірностей.

### 3.4 Формалізація та проектування системи підтримки прийняття рішень

Формалізація задачі на рівні поверхневого проектування зображена у вигляді діаграми варіантів використання. Різновиди використання програми зображені на рис. 3.4.

Як можна спостерігати з діаграми, інтернет-користувач при вході в програму мусить зареєструватися або авторизуватися, відвідувач може спостерігати доступні технології формування backend (хмарну чи виділену), для них передбачене індивідуальне меню з даними, робити запити на пошук з тестових даних, запит на відправку або отримання інформації на обраний сервер, формувати графік відповідностей та інші можливості, які представлено далі на схемі.



Рис. 3.4. Можливості налаштування системи

Основна задача бази даних – безпечне збереження великих обсягів інформації (тобто записи даних) та отримання доступу до бази користувачу або ж прикладному програмному забезпеченню.

База даних системи повинна містити тестові дані для роботи з сервером на прикладі інформації про місця харчування, дані про відвідувача, список друзів відвідувача, список замовлень відвідувача. База даних має зберігатись як на хмарному сервері, так і на виділеному, в ідентичному вигляді. Заклади повинні мати меню страв з ціною і коротким описом.

Визначення різновидів сутностей та їхніх зв'язків показує концептуальна модель бази даних. На цьому рівні визначаються ключові атрибути, атрибути, сутності, об'єкти, зв'язки між ними.

Логічне проектування – це процестрансформації концептуального образу бази даних в логічну структуру, включаючи формування зв'язків.

Дії, що здійснюються на цьому етапі проектування, відповідають обраній моделі даних. Трансформація від концептуальної до логічної моделі, для реляційної моделі, формується наступним чином:

1. приведення ER-діаграми до вигляду, прийнятого реляційною моделлю: даний крок полягає у вилученні рекурсивних і двохсторонніх зв'язків «багато до багатьох», вилученні багатозначних атрибутів, вилученні складних зв'язків. В даній концептуальній моделі таких зв'язків немає;

2. Відтворення концептуальної моделі на логічну, а саме формулювання набору таблиць (відносин), зв'язків між таблицями та ключових полів: здебільшого, сутність перетворюється на таблицю, первинний ключ на первинний ключ таблиці; основна сутність надає підлеглій сутності клонії первинного ключа для застосування його як зовнішнього ключа;

3. контроль відносин правилами нормалізації на збіг з першими трьома формами нормалізації:

- відношення відповідає першій формі нормалізації, якщо воно не має повторюваних груп (усі його елементи є атомарним);
- відношення відповідає другій формі нормалізації, якщо воно належить першій формі нормалізації першій формі нормалізації першій формі нормалізації кожен його не первинний атрибут повністю залежать від первинного. (часткові залежності видалені);

- відношення відповідає третій формі нормалізації, якщо воно належить другій формі нормалізації і всі його не первинні атрибути нетранзитивно залежать від первинного ключа (транзитивні залежності видалені);

4. перевірки відповідності зв'язків вимогам користувачьких транзакцій;

5. формування вимог підтримки повноти інформації; обов'язковість даних, межа для доменів атрибутів, повнота сутностей, посилавна цілісність, користувачькі обмеження.

Так як вибір унікального ключа сутності формується з урахуванням загальної довжини атрибутів, мінімального числа необхідних атрибутів в ключі і присутності гарантій унікальності значень в даний момент часу і в недалекому майбутньому, то в якості первинних ключів для усіх сутностей потрібно додати поле ID – унікальний ідентифікатор.

В результаті формування логічної моделі отримаємо логічну схему БД. Виконаємо розробку попередньої логічної схеми БД, на основі зображення концептуальної схеми моделі даних. Приведемо результати розробки на рис. 3.5. При логічному проектуванні бази даних «Заклади харчування» були виконані наступні дії:

- виключені особливості, які не сумісні з реляційною моделлю даних, а саме – зв'язки «Багато до багатьох».
- відображена концептуальна схема на логічній – визначено набори відношень згідно логічної моделі, які приведено у таблиці.
- нормалізовано відношення БД до нормальної форми.
- уточнено первинні і зовнішні ключі.
- визначено вимоги підтримки цілісності даних.
- побудована логічна схема моделі – визначені набори відношень згідно логічної моделі.

Структура БД враховує типові документи, які формуються на харчовому підприємстві, містить дані для розрахунку собівартості продукції і відповідає типовим рішенням для структури даних харчового підприємства.

Схема бази даних – її структура, описана на формальній мові, що підтримується системою управління базами даних. В реляційних базах даних схема визначає таблиці, поля в кожній таблиці, а також відношення між полями і таблицями.

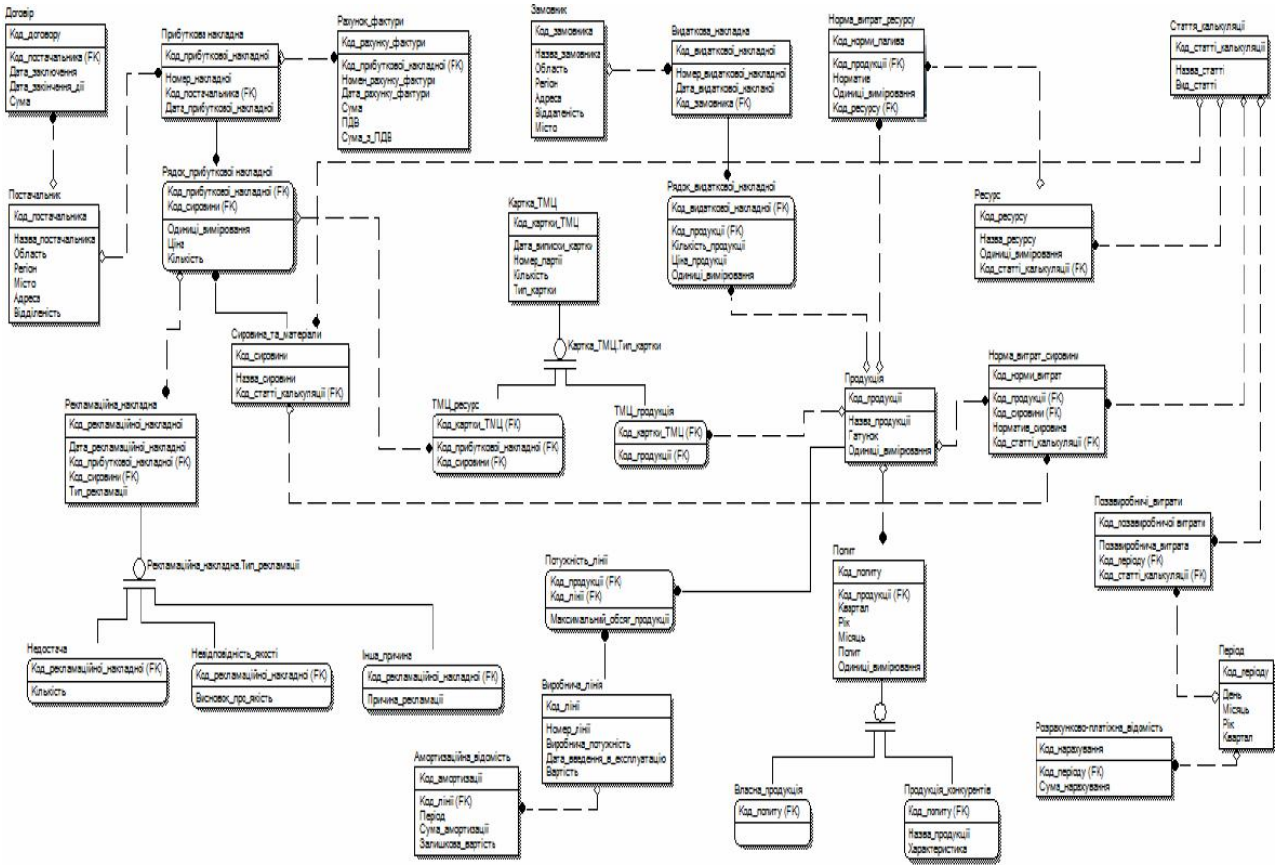


Рис. 3.5. Логічна схема бази даних

Схема бази даних була розроблена в середовищі Microsoft SQL Server Management Studio.

### 3.5 Підготовка управлінських рішень на основі аналізу даних OLAP та методами DataMining

Розглянемо на основі даних гіперкуба "Собівартість продукції в розрізі постачальників" (рис. 3.6).

Підприємство працює з деякими постачальниками цукрової пудри. Для практичної реалізації алгоритму щодо укладання договорів постачання на наступний рік необхідно:

1. Завантажуємо гіперкуб до OLAP-клієнта MS Excel і фільтруємо його по назві сировини – цукрова пудра.

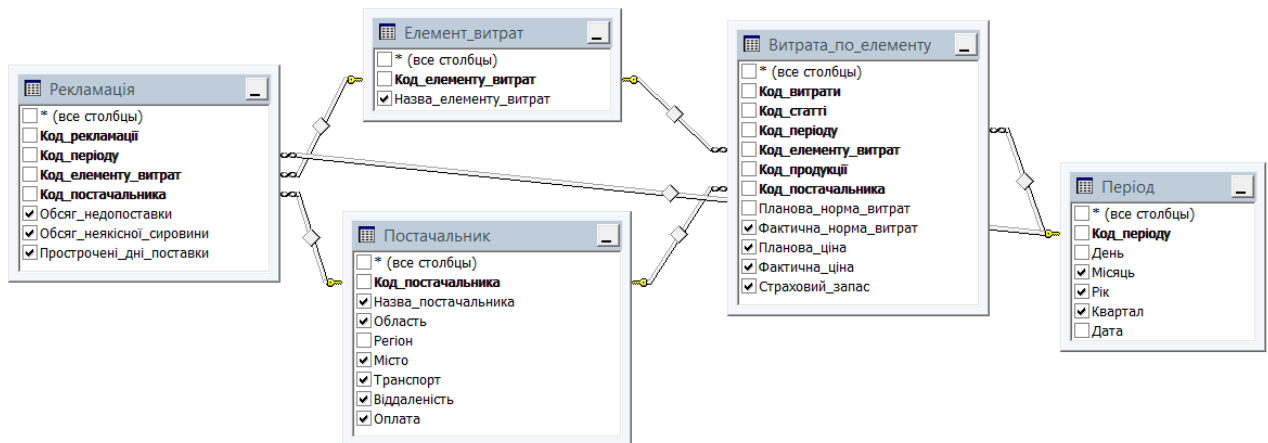


Рис. 3.6. Представлення OLAP-куба "Собівартість продукції в розрізі постачальників"

2. Використовуючи питання у вузлах дерева рішень, на кожному кроці фільтруємо підмножини постачальників ресурсів згідно питань вузлів дерева рішень для отримання рекомендацій щодо заключення договору на постачання ресурсу:

- формуємо підмножину постачальників, які влаштовують ОПР по критерію якості ресурсу;
- з обраної на попередньому кроці підмножини формуємо підмножину постачальників, які влаштовують ОПР по критерію ціни ресурсу;
- на наступному кроці формуємо множину постачальників, які виконують графік постачання;
- для сформованої підмножини формуємо підмножину постачальників, які можуть забезпечити потреби підприємства в обраному виді сировини або
- забезпечують страховий запас.

Детальний аналіз діяльності кожного постачальника дозволив зазначити, що найбільш пріоритетною є робота з постачальником "Зірка", "Літо", які забезпечують найкраще співвідношення по ціні та обсягам поставки. Таким чином, ОПР може рекомендувати укладання договорів з обраними постачальниками, зазначивши пріоритети співпраці з ними.

З метою пошуку шляхів зниження витрат використаємо алгоритм на основі методів Data Mining для виробітки пропозицій щодо коригування структури асортименту продукції, бо при плануванні виготовлення продукції на майбутні періоди виникає питання, яка продукція буде мати найбільший попит, чи є залежність між рецептурою продукції, її вагою, іншими характеристиками та збутом.

Пошук прихованих закономірностей здійснюється на основі OLAP-куба "Витрати та реалізація продукції", переданого до OLAP-клієнта MS Excel з надбудовою "Клієнт інтелектуального аналізу даних". Використання даної надбудови забезпечує можливість застосування алгоритмів служб MS Analysis Services в середовищі MS Office.

Спрогнозувавши попит на продукцію групи А та оцінивши резерви потужностей, ОПР може планувати заходи, пов'язані зі збільшенням обсягів її виробництва. Використавши групу А, виявимо характеристики продукції найбільш привабливої для споживача, застосувавши алгоритм дерева рішень.

Для проведення аналізу впливу характеристик продукції на обсяги її реалізації було виділено наступні характеристики: вага продукції, термін придатності, ціна, наявність позначки "Без ГМО", тип начинки, тип тіста. Дані характеристики враховують класифікацію характеристик і особливості кондитерського та хлібопекарського виробництва.

За рахунок використання алгоритму інтелектуального аналізу Microsoft Decision Trees, обмежено набір значущих характеристик, та визначено ті, які найбільш суттєво впливають на привабливість продукції це вага, тип начинки (рис. 3.7), а сформоване алгоритмом дерево рішень дозволило визначити значення характеристик найбільш привабливої продукції.

Аналізуючи побудоване дерево рішень, можна зробити висновок, що найбільш привабливою серед тих, що виготовляються, є продукція з шоколадним кремом, вагою більше 100 г. В меншому обсязі реалізується продукція з абрикосовим кремом.

Якщо врахувати, що методи кластеризації та дерева рішень виявили потенційно привабливі види продукції груп В та С, то доцільно спрогнозувати її собівартість та обсяги реалізації на майбутні періоди. Визначення собівартості продукції здійснено методом екстраполяції, методом кореляційно-регресивних моделей, а також на основі використання алгоритмів інтелектуального аналізу даних Data Mining. Обсяг реалізації спрогнозовано методом часових рядів Data Mining.

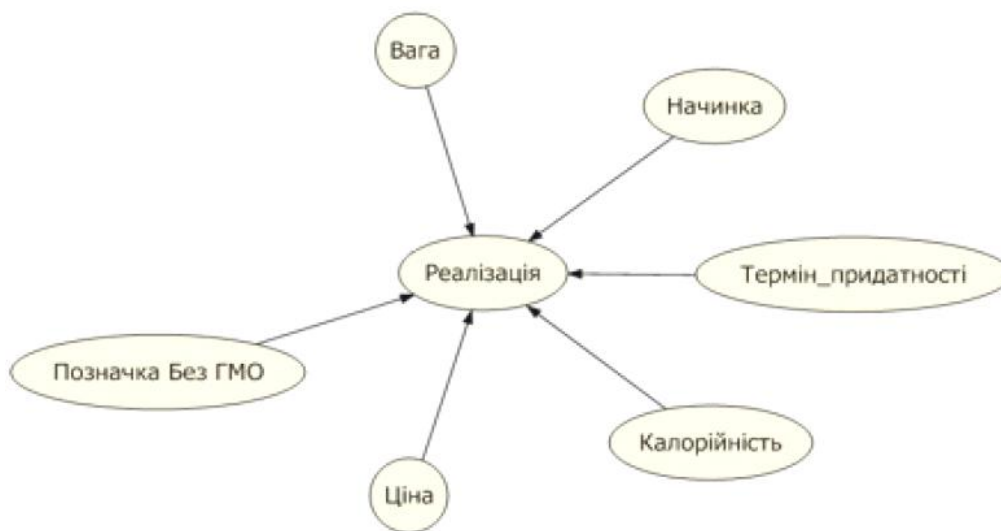


Рис. 3.7. Характеристики найбільш привабливої продукції

Для аналізу закономірностей та підготовки прогнозів при реалізації СППР використано алгоритм часових рядів (Майкрософт), який являє собою сукупність алгоритмів ARTXP та ARIMA. Алгоритм ARTXP використовує модель дерева з авторегресією та оптимізований для прогнозування наступного значення ряду. Даний алгоритм використовується для короткострокового прогнозування. Алгоритм ARIMA використовує інтегровані ковзні середні авторегресії і призначений для довгострокового прогнозування. Алгоритми

підтримують автоматичне виявлення сезонності або періодичності у даних на основі швидкого перетворення Фур'є, а також дозволяють вручну задати значення параметру сезонності.

Спрогнозуємо попит на продукцію на наступний період засобами Data Mining. Зображена на рис. 3.8 крива відображує обсяг реалізації продукції за попередній рік і показує прогноз на наступний місяць. Як видно з графіку, найкраще реалізується продукція з шоколадом в 1 та 12 місяці, а найгірше 6,7,8 місяцях. Прогноз обсягу реалізації на 13 місяць становитиме 5289 одиниць.

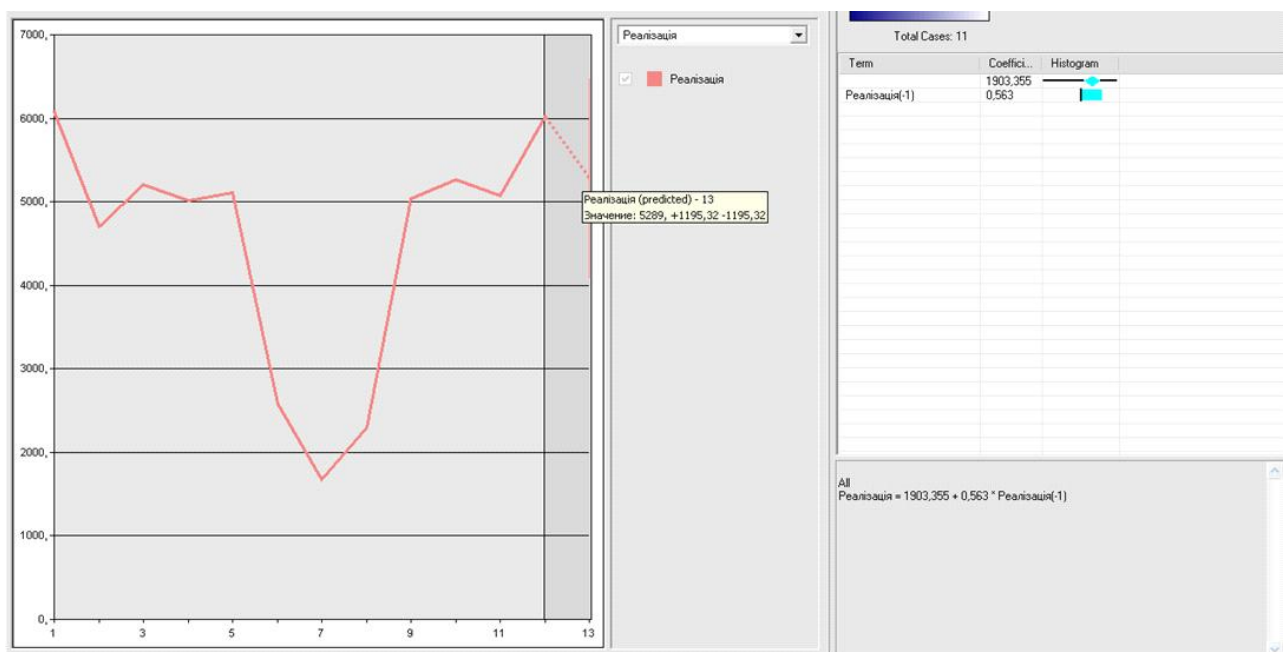


Рис. 3.8. Прогнозування реалізації продукції засобами Data Mining

Таким чином, в результаті реалізації етапів задачі коригування асортименту продукції засобами Data Mining визначено:

- найбільш прибуткові види продукції;
- сукупності характеристик, які впливають на привабливість продукції;
- потенційно привабливі види продукції зі знайденими сукупностями характеристик серед продукції, яка приносить меншу частку прибутку;

- продукцію зі знайденими сукупностями характеристик, мінімальними витратами на сировину, що дорого коштує, та максимальним попитом;
- спрогнозовано собівартість знайденої продукції з оцінкою ризику прийняття її рівня;
- спрогнозовано попит на продукцію;
- здійснено оцінку прийняття прогнозного рівня собівартості та попиту з використанням графічного та аналітичного методів обчислення беззбитковості.

Результати роботи алгоритму коригування асортименту продукції необхідно враховувати при плануванні заходів щодо зниження витрат:

- при плануванні заходів по збільшенню обсягів виготовлення продукції;
- плануванні нових видів продукції;
- необхідності заміни сировини.

### **3.6 Опис та програмна реалізація системи підтримки прийняття рішень**

Через те, що мова програмування Java працює на віртуальній машині, вона може ділитися програмними інтерфейсами. У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

Графічний інтерфейс в операційній системі Android представляє собою ієрархією об'єктів `android.view.View` і `android.view.ViewGroup`. Кожен об'єкт `ViewGroup` представляє контейнер, який містить і впорядковує дочірні об'єкти `View`. Об'єкти `View` є елементами управління та інші віджети, наприклад, кнопки, текстові поля і т.д., через які користувач взаємодіє з програмою, рис. 3.9.

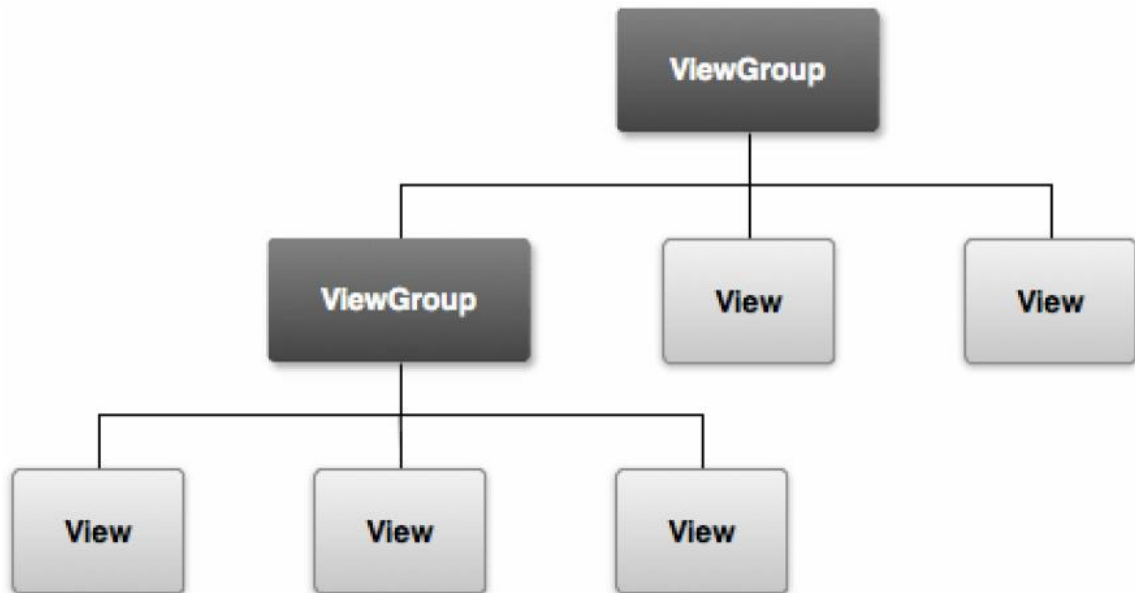


Рис. 3.9. Ієрархія об'єктів android.view.View

Оснoву обміну інформацією користувача із системою складають діалоги.

Інтерфейс програми має бути зручним та інтуїтивно зрозумілим, не викликати роздратованості. Ніякі дії користувача не повинні призвести до помилок в системі, у разі виникнення непередбачуваних станів, користувач повинен отримати вичерпне повідомлення про помилку, можливі причини її виникнення та рекомендовані подальші дії для її усунення.

Для того, щоб мати уявлення про інтерфейс користувача на етапі проектування були розроблені макети інтерфейсу, за допомогою дизайнерської платформи «UxPin» які наведені на рис. 3.10.

На рис. 3.10 схематично наведено три вікна додатка. На першому макеті наведено вікно входу у систему, можна побачити текстове поле TextView з підказкою для входу, два поля для введення інформації (логіна та пароля) EditText, та дві кнопки Button (для входу та створення акаунту). На наступному макеті зображено дві вкладки TabView, на першій повинна бути мапа даних, які отримані після виконання всіх запитів у системі, на іншій наведено перелік серверів за допомогою компоненту ListView. Також внизу розміщено кнопку Button для додання серверу до списку обраного. На останньому макеті наведено детальний опис обраного серверу, який складає: зображення марки серверу

ImageView, назва серверу, та коротка інформація. Перелік використаних компонентів з їх описом та властивостями наведено у таблиці 3.1.

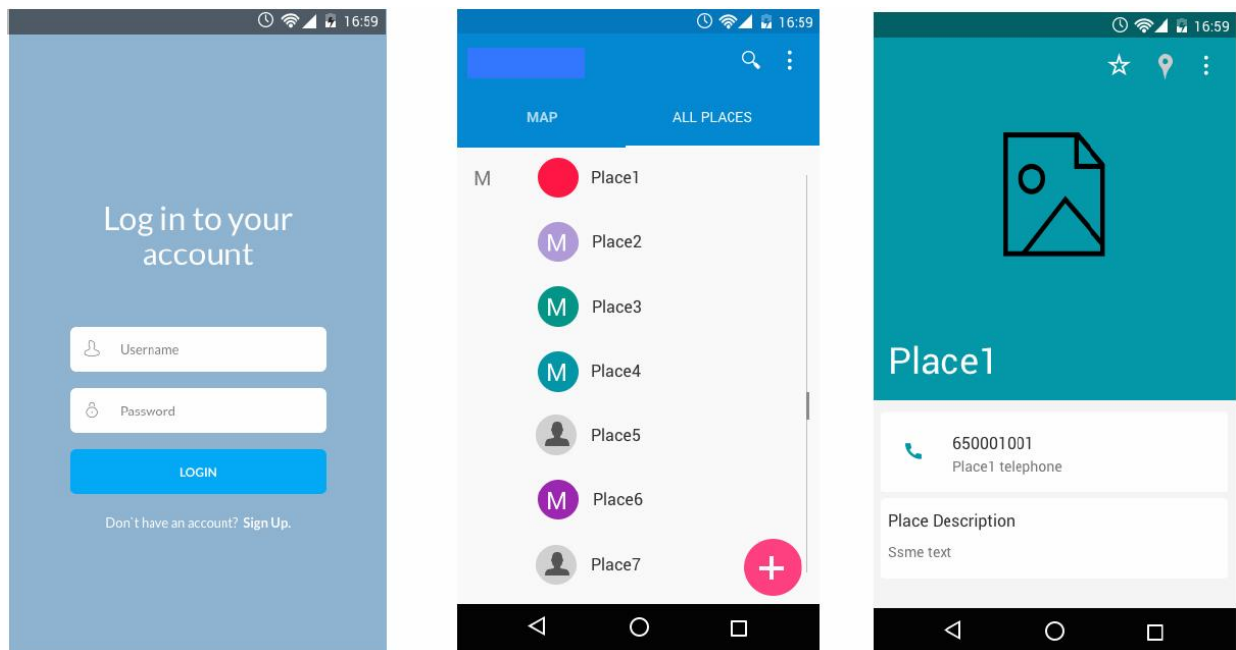


Рис. 3.10. Макети інтерфейсу

Таблиця 3.1.

Перелік використаних компонентів.

Назва компоненту	Опис	Використані властивості
1	2	3
ImageView	Призначений для перегляду зображення.	layout_width="w_content" layout_height="w_content" src="@drawable/my_comps"
Button	Кнопка, відповідає на натиснення, або іншу подію.	text="@string/button" onClick="sendMessage"
ScrollView	Макет контейнера для перегляду ієрархії, можна прокручувати користувачем, що дозволяє йому бути більше, ніж фізичною частиною дисплея.	layout_width="match_parent" layout_height="match_parent"
EditText	Текстове поле з можливістю введення інформації.	layout_weight="1" layout_width="0dp" layout_height="w_content" hint="@string/e_message"
RelativeLayout	Представляє об'єкт ViewGroup, який має в своєму розпорядженні дочірні елементи відповідно до їх координат.	paddingLeft="16dp" paddingRight="16dp"
TextView	Відображає текст без можливості редагування.	text="@st/display_mess" layout_centerHorizontal="true" textSize="16sp"

Переглянемо екран програми, який ілюструє інтерфейс системи при вході або реєстрації користувача на рис. 3.11.

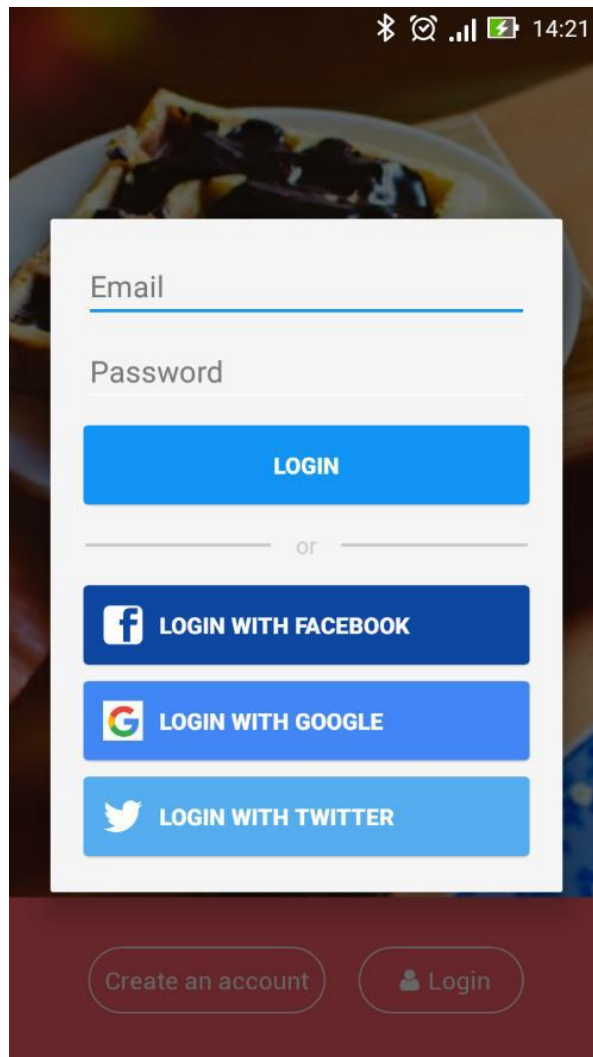


Рис. 3.11. Вхід в систему – вікно авторизації

На рис. 3.11 наведено вікно входу до програми, користувач може обрати створення нового облікового запису, або увійти до існуючого, на задньому плані можна побачити слайдер з коротким описом додатку. При перелистуванні зображення та написи опису змінюються. Якщо користувач обрав вхід до існуючого облікового запису він може обрати декілька типів авторизації.

Коли користувач увійшов до облікового запису, перед ним з'являється головна вкладка, рис. 3.12, на якій зображено список доступних серверів, на цьому екрані користувач може обрати доступний сервер та виконати на ньому один з запитів, для отримання даних та інформації про технологію його розробки.

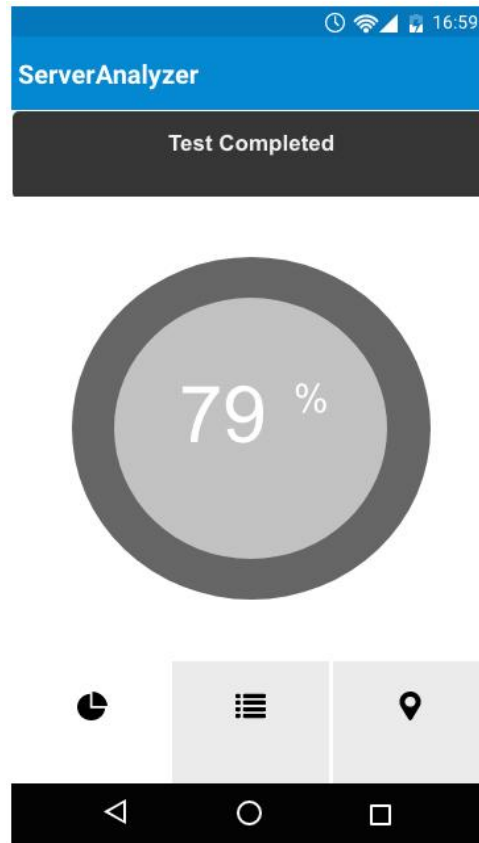


Рис. 3.12. Робота системи – вікно виконання запиту

Розроблена система на основі використання OLAP рішень та адаптованих алгоритмів інтелектуального аналізу Data Mining надасть можливість отримання, об'єднання та коригування даних з метою формування багатовимірного масиву інформації для підтримки прийняття управлінських рішень.

### 3.7 Висновки до розділу

Реалізації системи підтримки прийняття рішень для вирішення задач планування багатономенклатурного харчового підприємства передбачає наступні етапи:

1. Проектування схеми сховища даних з урахуванням всіх параметрів даних для аналізу собівартості продукції.
2. Проектування та побудова OLAP-кубів, наповнених інформацією, необхідною для аналізу факторів, які впливають на собівартість продукції.

3. Підготовка даних для формування прогнозних рішень на основі алгоритмів інтелектуального аналізу даних Data Mining.

4. Розробка рішень щодо візуалізації даних користувачеві в OLAP-клієнті для інформаційної підтримки прийняття управлінських рішень планування.

Отже, впровадження СППР на основі сховища даних та технології Data mining дозволить скоротити інтервали переоцінки асортименту до щомісячних за рахунок автоматизації процесу інтеграції даних. Сховище даних також підвищить достовірність і обсяг статистичної інформації, прискорить застосування управлінського впливу. Результатом впровадження СППР стане збільшення доходів підприємства, скорочення збитків за окремими напрямками за рахунок виявлення помилок на ранніх стадіях.

## ВИСНОВКИ

В магістерській роботі виконано дослідження моделей, методів та алгоритмів процесів підтримки прийняття рішень основі технологій OLAP і Data Mining. Значення результатів роботи полягає в формуванні інформаційної системи для підвищення ефективності діяльності підприємства за рахунок інформаційної підтримки прийняття управлінських рішень в задачах планування засобами Data Mining.

В роботі було проведено аналіз і дослідження розроблення інформаційних систем, розглянули основні етапи реалізації системи підтримки прийняття рішень для вирішення задач планування харчового підприємства, систем на основі онтологій.

Здійснено дослідження алгоритму оперативного управління собівартістю продукції на основі використання OLAP і Data Mining технологій для вирішення задач моніторингу, планування та зниження витрат харчового підприємства.

Здійснено розробку інформаційної системи аналізу структури асортименту продукції засобами аналізу даних Data Mining для інформаційної підтримки підготовки управлінських рішень щодо зниження витрат на виготовлення продукції.

В результаті виконаної магістерської роботи, було спроектовано та розроблено додаток для інформаційної підтримки прийняття рішень в задачах планування виробництва харчових підприємств. Спроекований програмний продукт складається з двох частин: серверна частина містить БД для переліку та надання тестових даних, логів тощо, а клієнтська – має інтерактивний інтерфейс для перегляду списку виконання запитів на них, складання графіків та виконання аналізу результатів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Овчарук В. О. Концепція вивчення ринку в системі автоматизованого управління якістю продукції [Текст] / В. О. Овчарук, Л. В. Зоткіна, О. В. Харкянен // Харчова промисловість. – 2001. – № 1. – С. 74–77 .
2. М'якшило О. М. Багатовимірні моделі даних в задачах планування діяльності харчового виробництва [Текст] / О. М. М'якшило, О. В. Харкянен // Харчова промисловість. – 2007. – №5. – С. 97-99
3. Демиденко М.А. Управління проектами інформатизації. /Навч. посібн. – Дніпропетровськ: НГУ, 2014. – 118 с.
4. Архіпова, Т. Л., and В. В. Крініцин. "Економічна оцінка ефективності інформаційних систем." Науковий вісник КУЕІТУ 2 (2011).
5. Power D.J. A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, [http:// DSSResources.COM/history /dsshhistory.html](http://DSSResources.COM/history/dsshhistory.html), version 4.1, May 31, 2019.
6. Bonczek, R., C. Holsapple and A. Whinston, Foundations of Decision Support Systems. New York: Academic, 1981.
7. George M. Marakas, Decision Support Systems in the 21st Century, 2nd Edition, Prentice Hall, 2002
8. Моделі і методи прийняття рішень в аналізі та аудиті: [навчальний посібник для студентів спец. 7.050106 «Облік і аудит»] / Ф. Ф. Бутинця, М. М. Шитун. — Житомир :ЖДГУ, 2004. — 352 с.
9. Гнатієнко Г.М. Експертні технології прийняття рішень / Гнатієнко Г.М., Снитюк В.Є. – К.: ТОВ „Маклаут”, – 2008. – 444 с.
10. Верес О. М. Компоненти концептуальної моделі системи підтримки прийняття рішень / О. М. Верес // Вісн. Нац. ун-ту “Львівська політехніка”. Серія : Комп’ютерні науки та інформаційні технології. – 2010. – № 686. – С. 103–112.
11. Гавриленко В. В. Комп’ютерні технології обробки економічної інформації : навч. посіб. / Гавриленко В. В., Парохненко Л. М. ; М-во освіти і науки України, Нац. трансп. ун-т. — К. : НТУ, 2008. — 187 с.

12. Бідюк П.І. Система підтримки прийняття рішень для аналізу фінансових даних [Текст] / Бідюк П.І., Кузнєцова Н.В., Терентьєв О.М. // Наукові вісті НТУУ “КПІ” – №1. –К.: НТУУ “КПІ” ВПІ ВПК “Політехніка”, 2011. – С. 48-61/

13. Parsaye K. OLAP and Data Mining/ Parsaye K. // Database Programming and Design. 1997. – № 2. 3.

14. Shakhovska N. B. Big Data Federated Repository Model / N. B. Shakhovska, Yu. Ja. Bolubash, O. M. Veres // (CADMS’2015) Proc. of the XIII Int. Conf., (Polyana-Svalyava (Zakarpattya), Ukraine, 24–27 February, 2015). – Lviv: Publishing Lviv Polytechnic, 2015. – P. 382–384.

15. Д. С. Корнеєв, Використання апарату нейронних мереж для створення моделі оцінки та управління ризиками підприємства. Київ: "Університет банківської справи", 2007.

16. Інтелектуальні інформаційні системи в умовах невизначеності і ризику [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/6020505/page:26/>.

17. A comparative analysis of multilevel computer-as-sisted decision making systems for traumatic injuries / S.Y. Ji [et al.] // BMC Med Inform Decis Mak. – 2009. Jan 14. – Vol. 9, N 2. – P. 1–17

18. V.P. Osipov, T.V. Sivakova, V.A. Sudakov, and Yu.I. Nechaev, “Intellectual core of the decision support system”, Preprints of the IPM named after M.V. Keldysh, no. 205, 23 p., 2018. doi: 10.20948/prepr-2018-205.

19. V.P. Karelin, “Intellectual technologies and artificial intelligence systems for decision support”, Bulletin of the Taganrog Institute of Management and Economics, no. 2, pp. 79–84, 2011. 6.E.A. Engel, “Models and methods of intellectual support for manager decision making”, Journal of Siberian state University of science and technology, no. 4, pp. 106–112, 2011.

20. O. Tikhanychev, “About information support of decision-making support”, International Journal “Software Products and Systems”, vol. 27, pp. 311–315, 2018. doi: 10.15827/0236-235X.122.311-315 8.R. Bavad and Jean-Charles Pomerol, “An “Intelligent” DSS for the Reinforcement of Urban Electrical

Power Networks”, Conference: Decision Support Systems: Experiences and Expectations, Proceedings of the IFIP TC8/WG8.3 Working Conference on Decision Support Systems: Experiences and Expectations, Fontainebleau, France, 30 June – 3 July 1992, pp. 153–165.

21. G. Phillips-Wren, “Decision Support Systems”, in book: Multicriteria Decision Aid and Artificial Intelligence. Loyola University Maryland, 2013, pp. 25–44. doi: 10.1002/9781118522516.ch2.

22. A. Kaklauskas, “Intelligent Decision Support Systems. In book: Biometric and Intelligent Decision Making Support”, Intelligent Systems Reference Library. Vilnius Gediminas Technical University, 2015, pp. 31–85. DOI: 10.1007/978-3-319-13659-2\_2.

23. V.P., Karelin and O.L. Kuzmenko, “Choosing the best management solution with fuzzy source data and multiple criteria”, Izvestiya VUZov. The North Caucasus region. Technical sciences, no.1, pp. 158–165, 2006.

24. A.A. Evseeva, D.A. Krasnikova, and A.A. Kazakov, “Modern approaches to solving multi-criteria problems”, Scientific and methodological electronic Journal “Concept”, vol. 3, pp. 2801–2805, 2013.

25. A. Stenin, I. Drozdovych, and M. Soldatova, “Situational management of urban engineering networks with intelligent support for dispatching decisions”, Proceedings of the Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020) Zaporizhzhia, Ukraine, April 27 – May 1, 2020, pp. 118–131. Available: <http://ceur-ws.org/Vol-2608>

26. M.L. Minsky, “A Framework for Representing Knowledge”, in Patrick Henry Winston (ed.), The Psychology of Computer Vision. New York: McGraw-Hill, 1975, pp. 211–277.

27. M.K. Hasan, “A Framework for Intelligent Decision Support System for Traffic Congestion Management System”, Engineering, Kuwait University, vol. 2, pp. 270–289, 2010. doi: 10.4236/eng.2010.24037

28. A.A. Akhrem, M.R. Ashinyants, and S.A. Petrov, “Fuzzy logical conclusion in the decision-making system”, Proceedings of the Institute of System Analysis of the Russian Academy of Sciences, vol. 29, pp. 265–275, 2007.

29. Azadeh, A., Sharifi, S., Saberi, M. (2009). Design and Implementation of a Human Centered Expert System for Improvement of Strategic Planning in a Manufacturer of Construction Products. *Australian Journal of Basic and Applied Sciences*, 3, 3, 2447-2458.
30. Burstein, F., Holsapple, C. W. (2008). *Handbook on Decision Support Systems 1: Basic Themes*. Springer Berlin Heidelberg.
31. 3. Burstein, F., Holsapple, C. W. (2008). *Handbook on Decision Support Systems 2: Variations*. Springer Berlin Heidelberg.
32. Fitzgerald, S. (2002). *Decision making*. Oxford: Capstone Publishing.
33. Gelinas, U., Sutton, S., Fedorowicz, J. (2004). *Business processes and information technology*. Cincinnati: South-Western/Thomson Learning.
34. Inmon, W. (2005). *Building the data warehouse (4th ed.)*. Indianapolis: Wiley.
35. Ivanov, D., Tsipoulanidis, A., Schönberger Jörn. (2019). *Global supply chain and operations management: a decision-oriented introduction to the creation of value*. Cham, Switzerland: Springer Nature Switzerland AG.
36. March, J. G., Simon, H. A. (1958). *Organizations*. New York: Wiley & Sons.
37. Marcomini, A., Suter II, G. W., Critto, A. (Eds.). (2008). *Decision support systems for risk-based management of contaminated sites*. Springer Science & Business Media.
38. Parsaei, H. R. (1996). *Manufacturing decision support systems*. Boston, MA: Springer US.
39. Sauter, V. (2010). *Decision support systems for business intelligence (2nd ed.)*. Hoboken, N. J: Wiley & Sons.
40. Turban, E., Aronson, J., Liang, T., Sharda, R., Turban, E. (2004). *Decision support and business intelligence systems*. Prentice Hall. [in English].
41. Guzhva V. M. (2001). *Informatsijni systemy I tehnologiji na pidpryjemstvah: navchalnyj posibnyk [Information systems and technologies at enterprises: the manual]*. Kyiv: KNEU.

42. Aditya Das. (2013). Artificial intelligence and decision support systems. Retrieved February 02, 2020, from <https://prezi.com/knlzigzxme9a/artificial-intelligence-and-decision-support-systems/>.
43. Herre, H. (2010). General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In *Theory and applications of ontology: computer applications* (c. 297–345). Springer, Dordrecht.
44. Raz et al. (2006). *Fast and Efficient Context-Aware Services*. John Wiley & Sons.
45. Prasenjit, M., & Gio, W. (2003). An Ontology-Composition Algebra
46. Tarapata, Z. (2007). Multicriteria weighted graphs similarity and its application for decision situation pattern matching problem. In *Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics* (c. 1149–1155).
47. Miah, S.J., Gammack, J. & Kerr, D., (2007). Ontology development for context-sensitive decision support. In *Semantics, Knowledge and Grid, Third International Conference on* (c. 475–478). IEEE.

# ДОДАТКИ

## Додаток А

### Програмна реалізація системи

#### Зміст файлу “build.gradle”

```

apply plugin: 'com.android.application' android {
//задаємо версії засобів розробки compileSdkVersion 23
    buildToolsVersion "23.0.3" defaultConfig {
        applicationId "com.example.koza4.ServerAnalizer"
        minSdkVersion 21
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    packagingOptions {
        exclude 'META-INF/LICENSE' exclude 'META-INF/NOTICE' exclude
        'META-INF/NOTICE.txt'
    }
}
//підключення зовнішніх бібліотек dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar']) testCompile
    'junit:junit:4.12'
    compile project(':myapplication')
    compile 'com.android.support:design:24.0.0-beta1' compile
    'com.android.support:cardview-v7:24.0.0-beta1' compile
    'com.github.bumptech.glide:glide:3.6.0'
    compile 'de.hdodenhof:circleimageview:1.3.0'
    compile 'com.android.support:appcompat-v7:24.0.0-beta1'
    compile 'com.flaviofaria:kenburnsview:1.0.6'
    compile 'com.google.android.gms:play-services-auth:9.0.2' compile
    'com.firebase:firebase-client-android:2.3.1' compile
    'com.android.support:recyclerview-v7:24.0.0-beta1' compile
    'com.facebook.android:facebook-android-sdk:4.6.0' compile
    'com.viewpagerindicator:library:2.4.1'
    compile 'com.github.medyo:fancybuttons:1.5@aar'
    compile 'com.google.android.gms:play-services:9.0.2'
    compile 'com.google.android.gms:play-services-ads:9.0.2' compile
    'com.google.android.gms:play-services-gcm:9.0.2'
}

```

#### Вміст файлу strings.xml:

```

<resources>
<!-- Задаємо значення строковим ресурсам додатку -->
    <string name="app_name">ServerAnalizer</string>
    <string name="splash_bg">qq</string>
    <string name="fontello_logo">&#xe800;</string>
    <string name="fontello_search">&#xe816;</string>
    <string name="fontello_x_mark">&#xe81f;</string>
    <string name="fontello_x_mark_masked">&#xe806;</string>

```

```

<string name="fontello_microfon">&#xe80f;</string>
<string name="fontello_user">&#xe81b;</string>
<string name="fontello_password">&#xe823;</string>
<string name="fontello_upload">&#xe81a;</string>
<string name="fontello_play">&#xe813;</string>
<string name="fontello_heart_empty">&#xe80b;</string>
<string name="fontello_heart_full">&#xe80a;</string>
<string name="fontello_drag_and_drop">&#xe821;</string>
<string name="welcome">Welcome!</string>
<string name="facebook_app_id">377505602442747</string>
<string name="twitter_app_key">vvvcQrekKC6dlr9FNhhxxW10K</string>
<string name="twitter_app_secret">hooyQHlrwELa6Lqh</string>
<string name="action_settings">Settings</string>
<string name="app_overview_1st_item_text"></string>
<string name="app_overview_2nd_item_text"></string>
<string name="app_overview_3rd_item_text"></string>
<string name="app_overview_4th_item_text"></string>
<string name="icon_user">&#xf007;</string>
<string name="icon_add">&#xf067;</string>
<string name="icon_readmore">&#xf178;</string>
<string name="icon_answer">&#xf095;</string>
<stringname="icon_mute">&#xf131;</string>
<stringname="icon_like">&#xf087;</string>
<string name="icon_share">&#xf064;</string>
<string name="icon_follow">&#xf005;</string>
<stringname="icon_android">&#xf17b;</string>
<stringname="icon_dropbox">&#xf16b;</string>
<string name="icon_soundcloud">&#xf1be;</string>
<string name="icon_spotify">&#xf1bc;</string>
<string name="icon_vine">&#xf1ca;</string>
<stringname="icon_twitter">&#xf099;</string>
<stringname="icon_tumblr">&#xf173;</string>
<string name="icon_stackoverflow">&#xf16c;</string>
<string name="icon_gplus">&#xf0d5;</string>
<string name="icon_facebook">&#xf09a;</string>
<string name="icon_entypo_facebook">&#59140;</string>
<stringname="icon_behance">&#xf1b4;</string>
<stringname="icon_dribble">&#xf17d;</string>
<stringname="icon_upload">&#xf0ee;</string>
<stringname="icon_config">&#xf085;</string>
<string name="icon_creditcard">&#xf09d;</string>
<string name="ghost_button_title">Ghost Button</string>
<string name="icon_envelope">|</string>
<string name="checkin">Checkin</string>
<string name="menu_settings">Settings</string>
<string name="title_activity_maps">Map</string>
</resources>

```

### Вміст файлу activity\_login.xml:

```
<!-- Задаємо розмітку для вікна входу у систему -->
```

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/tools"
android:layout_height="match_parent" android:layout_width="match_parent"
xmlns:fancy="http://schemas.android.com/apk/res-auto"
android:fillViewport="true"><FrameLayout
android:layout_width="match_parent"
android:layout_height="match_parent"><FrameLayout
android:layout_width="match_parent" android:layout_height="match_parent"
android:layout_marginBottom="95dp"><com.flaviofaria.kenburnsviiew.KenBurnsView
android:layout_width="match_parent"
android:layout_height="match_parent" android:id="@+id/icon_image1"
android:scaleType="centerCrop"/><com.flaviofaria.kenburnsviiew.KenBurnsView
android:layout_width="match_parent"
android:layout_height="match_parent" android:id="@+id/icon_image2"
android:scaleType="centerCrop"
/></FrameLayout><android.support.v4.view.ViewPager
android:layout_width="fill_parent" android:layout_height="fill_parent"
android:id="@+id/intro_view_pager"/><com.viewpagerindicator.CirclePageIndicator
android:id="@+id/indicator" android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal|bottom"
app:fillColor="@color/material_grey_800"
app:pageColor="@color/material_grey_400" app:radius="3dp"
app:strokeWidth="1dp" app:strokeColor="#ff838b"
android:layout_marginBottom="84dp" />
<LinearLayout android:layout_width="match_parent"
android:layout_height="wrap_content" android:background="#ff5f69"
android:gravity="center" android:paddingBottom="30dp"
android:paddingTop="30dp"
android:layout_gravity="center_vertical|bottom"><mehdi.sakout.fancybuttons.FancyButton
android:id="@+id/btn_create_account"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_marginRight="20dp" android:padding="10dp"
fancy:fb_borderColor="#FFFFFF" fancy:fb_borderWidth="1dp"
fancy:fb_defaultColor="#ff5f69" fancy:fb_focusColor="#ff838b"
fancy:fb_radius="30dp" fancy:fb_text="Create an account"
fancy:fb_textColor="#FFFFFF"
></mehdi.sakout.fancybuttons.FancyButton><mehdi.sakout.fancybuttons.FancyButton
android:id="@+id/btn_login" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:paddingBottom="10dp"
android:paddingLeft="20dp" android:paddingRight="20dp"
android:paddingTop="10dp" fancy:fb_borderColor="#FFFFFF"
fancy:fb_borderWidth="1dp" fancy:fb_defaultColor="#ff5f69"
fancy:fb_focusColor="#ff838b"
fancy:fb_fontIconResource="@string/icon_user"
fancy:fb_iconPosition="left" fancy:fb_radius="30dp"
fancy:fb_text="Login" fancy:fb_textColor="#FFFFFF"
/></LinearLayout></FrameLayout></RelativeLayout>

```

### Вміст файлу side\_naw\_bar.xml

```

<?xml version="1.0" encoding="utf-8"?><!-- Малюємо панель навігації -->
<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle"><gradient android:angle="135"
android:endColor="@color/colorPrimaryDark"
android:startColor="@color/colorPrimary" android:type="linear"/></shape>
package com.example.koza4.ServerAnalizer; import android.content.Intent;
import android.content.pm.ActivityInfo; import android.os.Bundle; import

```

```

android.support.annotation.NonNull; import
android.support.design.widget.NavigationView; import
android.support.v4.view.PagerAdapter; import
android.support.v4.view.ViewPager; import
android.support.v7.widget.RecyclerView; import android.util.Log; import
android.view.View; import android.view.ViewGroup; import
android.view.Window; import android.view.animation.Animation; import
android.view.animation.AnimationUtils; import android.widget.Button;
import android.widget.ImageView; import android.widget.TextView; import
android.widget.Toast; import com.androidquery.AQuery; import
com.example.koza4.myapplication.ui.auth.core.AuthProviderType; import
com.example.koza4.myapplication.ui.auth.core.FirebaseLoginBaseActivity;
import com.example.koza4.myapplication.ui.auth.core.FirebaseLoginError;
import com.firebase.client.AuthData; import
com.firebase.client.Firebase; import com.firebase.client.Query; import
com.google.firebase.auth.FirebaseAuth; import
com.google.firebase.auth.FirebaseUser; import
com.viewpagerindicator.CirclePageIndicator; import
com.viewpagerindicator.PageIndicator; import
mehdi.sakout.fancybuttons.FancyButton; public class LoginActivity
extends FirebaseLoginBaseActivity { private ViewPager viewPager; private
PageIndicator indicator; private ImageView topImage1; private ImageView
topImage2;
private int lastPage = 0; private int[] icons; private int[] messages;
private Firebase mRef; public static String TAG = "FirebaseUI.chat";
private String mName; private FirebaseAuth mAuth; private
FirebaseAuth.AuthStateListener mAuthListener; //функція, що з'являється
під час створення Activity @Override protected void onCreate(Bundle
savedInstanceState) { super.onCreate(savedInstanceState);
requestWindowFeature(Window.FEATURE_NO_TITLE); mAuth =
FirebaseAuth.getInstance();
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
Firebase.setAndroidContext(this); //зв'язуємось з сервером Firebase
setContentView(R.layout.activity_login); //mLoginButton = (Button)
findViewById(R.id.btn_login); FancyButton startMessagingButton =
(FancyButton) findViewById(R.id.btn_login);
startMessagingButton.setOnClickListener(new View.OnClickListener() {
@Override public void onClick(View view) { Runnable runnable = new
Runnable() { //створюємо новий потік public void run() {
showFirebaseLoginPrompt(); } }; Thread thread = new Thread(runnable);
thread.start(); } }); FancyButton CreateAccButton = (FancyButton)
findViewById(R.id.btn_create_account);
CreateAccButton.setOnClickListener(new View.OnClickListener() {
@Override public void onClick(View view) { Intent mainIntent = new
Intent(LoginActivity.this, ThreeActivity.class);
LoginActivity.this.startActivity(mainIntent); /* Finish splash activity
so user cant go back to it. */ LoginActivity.this.finish(); } }); mRef =
new Firebase("https://glaring-torch-7273.firebaseio.com"); AQuery aq =
new AQuery(this); //задаємо слайдер
aq.id(R.id.icon_image1).image(R.drawable.app_overview_1_buy); icons =
new int[]{ R.drawable.app_overview_1_buy,
R.drawable.app_overview_2_smartphone,
R.drawable.app_overview_3_save_money, R.drawable.app_overview_4_coupons
}; messages = new int[] { //повідомлення слайдера
R.string.app_overview_1st_item_text,
R.string.app_overview_2nd_item_text,
R.string.app_overview_3rd_item_text, R.string.app_overview_4th_item_text
}; viewPager = (ViewPager) findViewById(R.id.intro_view_pager);
topImage1 = (ImageView) findViewById(R.id.icon_image1); topImage2 =
(ImageView) findViewById(R.id.icon_image2); indicator =

```

```

(CirclePageIndicator) findViewById(R.id.indicator);
topImage2.setVisibility(View.GONE); viewPager.setAdapter(new
IntroAdapter()); viewPager.setPageMargin(0);
viewPager.setOffscreenPageLimit(1); indicator.setViewPager(viewPager);
indicator.setOnPageChangeListener(new ViewPager.OnPageChangeListener() {
@Override public void onPageScrolled(int position, float positionOffset,
int positionOffsetPixels) { //при прокручуванні сторінки if (lastPage !=
viewPager.getCurrentItem()) { lastPage = viewPager.getCurrentItem();
final ImageView fadeoutImage; final ImageView fadeinImage; if
(topImage1.getVisibility() == View.VISIBLE) { fadeoutImage = topImage1;
fadeinImage = topImage2; } else { fadeoutImage = topImage2; fadeinImage
= topImage1; } fadeinImage.bringToFront();
fadeinImage.setImageResource(icons[lastPage]);
fadeinImage.clearAnimation(); fadeoutImage.clearAnimation(); Animation
outAnimation = AnimationUtils.loadAnimation(LoginActivity.this,
R.anim.icon_anim_fade_out); outAnimation.setAnimationListener(new
Animation.AnimationListener() { @Override public void
onAnimationStart(Animation animation) { } @Override public void
onAnimationEnd(Animation animation) {
fadeoutImage.setVisibility(View.GONE); } @Override public void
onAnimationRepeat(Animation animation) { } }); //анімація при
перелистуванні Animation inAnimation =
AnimationUtils.loadAnimation(LoginActivity.this,
R.anim.icon_anim_fade_in); inAnimation.setAnimationListener(new
Animation.AnimationListener() { @Override public void
onAnimationStart(Animation animation) {
fadeinImage.setVisibility(View.VISIBLE); }
@Override public void onAnimationEnd(Animation animation) { } @Override
public void onAnimationRepeat(Animation animation) { } });
fadeoutImage.startAnimation(outAnimation);
fadeinImage.startAnimation(inAnimation); } } @Override public void
onPageSelected(int position) { } @Override public void
onPageScrollStateChanged(int state) { } }); mAuthListener = new
FirebaseAuth.AuthStateListener() { @Override public void
onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) { FirebaseUser
user = firebaseAuth.getCurrentUser(); if (user != null) { // User is
signed in Log.d(TAG, "onAuthStateChanged:signed_in:" + user.getId()); }
else { // User is signed out Log.d(TAG,
"onAuthStateChanged:signed_out"); } } }; private class IntroAdapter
extends PagerAdapter { //клас для зображення слайдера @Override public
int getCount() { return 4; } @Override public boolean
isViewFromObject(View view, Object object) { return view.equals(object);
} @Override public Object instantiateItem(ViewGroup container, int
position) { View view = View.inflate(container.getContext(),
R.layout.intro_view_layout, null); TextView messageTextView = (TextView)
view.findViewById(R.id.message_text); container.addView(view, 0);
messageTextView.setText(AndroidUtilities.replaceTags(getString(messages[
position]), getApplicationContext())); return view; }
@Override public void destroyItem(ViewGroup container, int position,
Object object) { ((ViewPager) container).removeView((View) object); } }
@Override protected void onStart() { super.onStart();
setEnabledAuthProvider(AuthProviderType.FACEBOOK);
setEnabledAuthProvider(AuthProviderType.TWITTER);
setEnabledAuthProvider(AuthProviderType.GOOGLE);
setEnabledAuthProvider(AuthProviderType.PASSWORD);
mAuth.addAuthStateListener(mAuthListener); } @Override public void
onStop() { super.onStop(); if (mAuthListener != null) {
mAuth.removeAuthStateListener(mAuthListener); } // ... } @Override //при
успішній авторизації public void onFirebaseLoggedIn(AuthData authData) {
Log.i(TAG, "Logged in to " + authData.getProvider().toString()); switch

```

```

(authData.getProvider()) { case "password": mName = (String)
authData.getProviderData().get("email"); break; default: mName =
(String) authData.getProviderData().get("displayName"); break; }
Toast.makeText(getApplicationContext(), mName,
Toast.LENGTH_LONG).show(); invalidateOptionsMenu(); Intent mainIntent =
new Intent(LoginActivity.this, ThreeActivity.class);
LoginActivity.this.startActivity(mainIntent); /* Finish splash activity
so user cant go back to it. */ LoginActivity.this.finish(); } @Override
public void onFirebaseLoggedOut() { //при виході з облікового запису
Log.i(TAG, "Logged out"); mName = ""; invalidateOptionsMenu(); }
@Override //при помилці входу public void
onFirebaseLoginProviderError(FirebaseLoginError firebaseError) {
Log.e(TAG, "Login provider error: " + firebaseError.toString());
resetFirebaseLoginPrompt();
} @Override //при помилці входу public void
onFirebaseLoginUserError(FirebaseLoginError firebaseError) { Log.e(TAG,
"Login user error: " + firebaseError.toString());
resetFirebaseLoginPrompt(); } @Override public Firebase getFirebaseRef()
{ return mRef; } }

```